COMPRESSIVE SAMPLING OF SPEECH SIGNALS

by

Mona Hussein Ramadan

BS, Sebha University, 2005

Submitted to the Graduate Faculty of

Swanson School of Engineering in partial fulfillment

of the requirements for the degree of

Master of Science

University of Pittsburgh

2010

UNIVERSITY OF PITTSBURGH

SWANSON SCHOOL OF ENGINEERING

This thesis was presented

by

Mona Hussein Ramadan

It was defended on

November 23, 2010

and approved by

Luis Chaparro, PhD, Associate Professor, Electrical Engineering

Patrick Loughlin, PhD, Professor, Bioengineering

Thesis Advisor: Amro El-Jaroudi, PhD, Associate Professor, Electrical Engineering

COMPRESSIVE SAMPLING OF SPEECH SIGNALS

Mona Hussein Ramadan, M.S.

University of Pittsburgh, 2010

Compressive sampling is an evolving technique that promises to effectively recover a sparse signal from far fewer measurements than its dimension. The compressive sampling theory assures almost an exact recovery of a sparse signal if the signal is sensed randomly where the number of the measurements taken is proportional to the sparsity level and a log factor of the signal dimension. Encouraged by this emerging technique, we study the application of compressive sampling to speech signals.

The speech signal is very dense in its natural domain; however speech residuals obtained from linear prediction analysis of speech are nearly sparse. We apply compressive sampling to speech signals, not directly but on the speech residuals obtained by conventional and robust linear prediction techniques. We use a random measurement matrix to acquire the data then use ℓ-1 minimization algorithms to recover the data. The recovered residuals are then used to synthesize the speech signal. It was found that the compressive sampling process successfully recovers speech recorded both in clean and noisy environments. We further show that the quality of the speech resulting from the compressed sampling process can be considerably enhanced by spectrally shaping the error spectrum. The recovered speech quality is said to be of high quality with SNR up to 15 dB at a compression factor of 0.4.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

PREFACE

I would first like to thank my advisor Dr. Amro El-Jaroudi for his constant support and guidance throughout my entire M.S. journey. I would also like to express my appreciation to my advisory committee members for their valuable time and feedback. My gratitude is extended to all my professors in the Department of Electrical Engineering for providing me with the knowledge that enabled me to pursue my degree.

I would also like to thank my family: Baba and Mama; and my brothers Mahmoud, Mostafa, Mumen and Mohamed for their trust, belief and support; and their consistent continuous love. This thesis is fully dedicated for them.

1.0 INTRODUCTION

Speech has always been the most popular tool of communication; speech processing has been an interesting field of study that attracted a lot of attention during the last 40 years. New technologies have been studied to reduce the speech transmission rates while maintaining a good quality of the transmitted speech. Compressive sampling is a new developing technique of data acquisition that offers a promise of recovering the data from a fewer number of measurements than the dimension of the signal. The goal of this work is to study and apply compressive sampling techniques on speech signals. We apply compressive sampling on speech residuals then synthesize the speech from the recovered residuals. The behavior of the recovered signals is thoroughly investigated for male and female speech signals recorded in both clean and noisy settings.

This document is divided into two parts. Part I is a background and literature review and is organized as follows. Chapter 2 provides an introduction to speech signals where the production mechanism and the classification of speech signals are briefly explained. In Chapter 3, some speech coding techniques are described. Linear prediction is explained in detail in Section 3.1. Since we apply compressive sampling to the residual signal, it is important to explain the linear prediction methods and the properties of the prediction filter and the prediction error. Section 3.2 highlights the multi-pulse excited linear prediction coding. The multi-pulse excitation is presented to get familiar with the sparse nature of the excitation signal and to introduce a pulse search algorithm that is comparable to the orthogonal matching pursuit algorithm presented later in Chapter 4. Robust linear prediction is presented in Section 3.3 since

it results in a prediction filter that better fits the speech spectrum. Compressive sampling is introduced in Chapter 4. The compressive sampling problem is stated and explained in detail; and examples are provided along with two possible solutions to the problem.

Implementation and result discussions are provided in Part II of this document. In Chapter 5, the compressive sampling process is applied to speech residuals obtained from conventional and robust linear prediction techniques and the recovery results are compared for the two cases. Chapter 6 addresses the spectral shaping of the compressive sensing noise. Spectral shaping as a concept is briefly introduced and several shaping filters are used to search for the filter that best shapes the noise and results in the best quality of speech.

The results of the implementation ,conclusions and future direction are summarized in Chapter 7.

2.0 THE SPEECH SIGNAL

Speech has always been the most dominant and common way of communication. The information contained in the spoken word is conveyed by the speech signal. In order to analyze speech transmission and processing, we need to understand the basic structure of the speech signal and its production models.

This chapter introduces the speech signal in an attempt to answer the questions of how speech is produced and how it could be modeled; what its main characteristics are and how it may be classified. Section 2.1 answers the first question, and Section 2.2 answers the last one.

2.1 HUMAN GENERATION OF SPEECH

The speech waveform is a sound pressure wave originating from controlled movements of anatomical structures making up the human speech production system [1]. Figure 1 shows a model of vowel production. In vowel production, air is forced from the lungs by contraction of the muscles around the lung cavity. Air then flows past the vocal cords, which are two masses of flesh, causing periodic vibration of the cords whose rate gives the pitch of the sound; the resulting periodic puffs of air act as an excitation input, or source, to the vocal tract. The vocal tract, which is the cavity between the vocal cords and the lips, acts as a resonator that spectrally shapes the periodic input.

A simple engineering model, referred to as the source/filter model, can thus be built based on this production mechanism. If we assume that the vocal tract is a linear time-invariant

system with a periodic impulse-like input, then the pressure output at the lips is the convolution of the impulse-like train with the vocal tract impulse response, and therefore is itself periodic [2]. This is a simple model of a steady-state vowel. The speech utterance consists of a string of vowel and consonant phonemes whose temporal and spectral characteristics change with time, corresponding to a changing excitation source and vocal tract system [2].

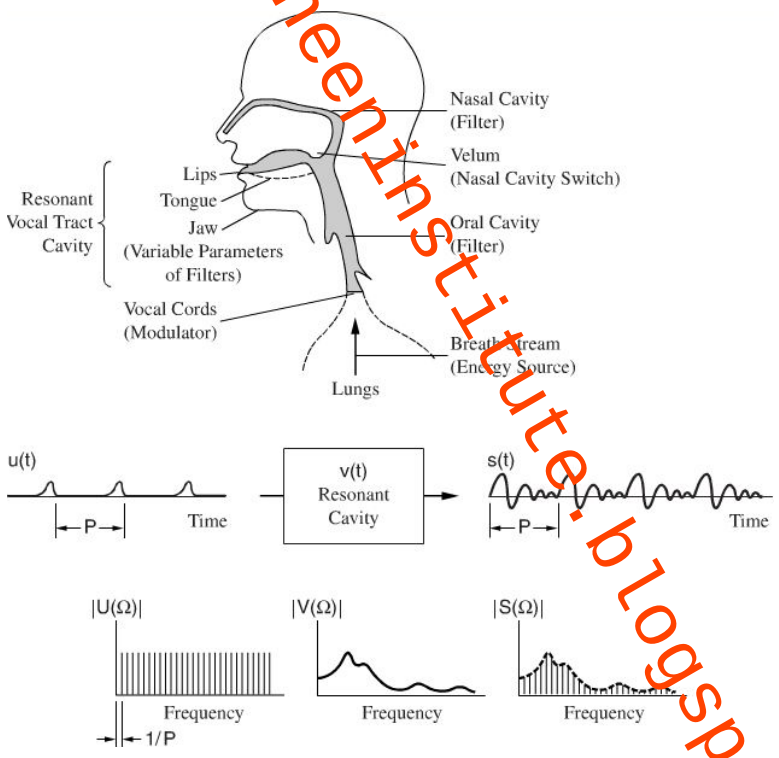

Figure 1. **Speech production mechanism and model of a steady-state vowel. The acoustic waveform is modeled as the output of a linear time-invariant system with a periodic impulse-like input. In the frequency domain, the vocal tract system function spectrally shapes the harmonic input [2].**

2.2 CLASSIFICATION OF SPEECH SIGNALS: VOICED VS UNVOICED

As described in Section 2.1, a sound source is generated by the vocal folds then spectrally shaped in the vocal tract to generate a sound. Sounds hence can be classified in many ways; either based

on the nature of the source (the air puffs) or the shape of the vocal tract (the position of the tongue and the degree of its constriction). Sounds can also be classified based on their time domain waveform or the time varying spectral characteristics [2]. Therefore, we need a specific classification of sounds that can be used in modeling the speech for digital signal processing applications.

Speech sounds can be roughly classified, based on the nature of the source, into voiced and unvoiced [3]. Voiced sounds are produced when air is forced through the vocal cords so their vibration results in a sequence of quasi-periodic pulses that excites the vocal tract. Unvoiced sounds result when forcing air through the vocal tract without vibrating the vocal cords [2].

Voiced and unvoiced sounds have different properties and hence are reproduced differently, as will be discussed in the next chapter. Therefore, it is important for some speech coders to classify the speech signal into voiced and unvoiced sounds. The main characteristics that are used to distinguish between voiced and unvoiced sounds are: periodicity, energy, and zero crossing.

2.2.1. Periodic nature of the speech signal

In the time domain, the voiced sound signal is clearly periodic with a fundamental frequency called the pitch. Pitch ranges from 50 to 250 Hz for men and from 120 to 500 Hz for women [1]. On the other hand, unvoiced sounds are not periodic and further have a random nature.

Figure 2 shows an example for a voiced and an unvoiced utterance, [oh] and [sh] respectively, by a female speaker and an expanded view of a 40 ms frame of each utterance. The expanded frame view shows the periodic nature of the voiced sound and the random nature of the unvoiced sound.

In the 40 ms slice of the voiced sound in Figure 2, the pattern repeats itself about nine times, where each repetition corresponds to one cycle of the vocal cords opening and closing. Thus the period of the pattern is about 4.44 ms and the fundamental frequency is then about 225.23 Hz.

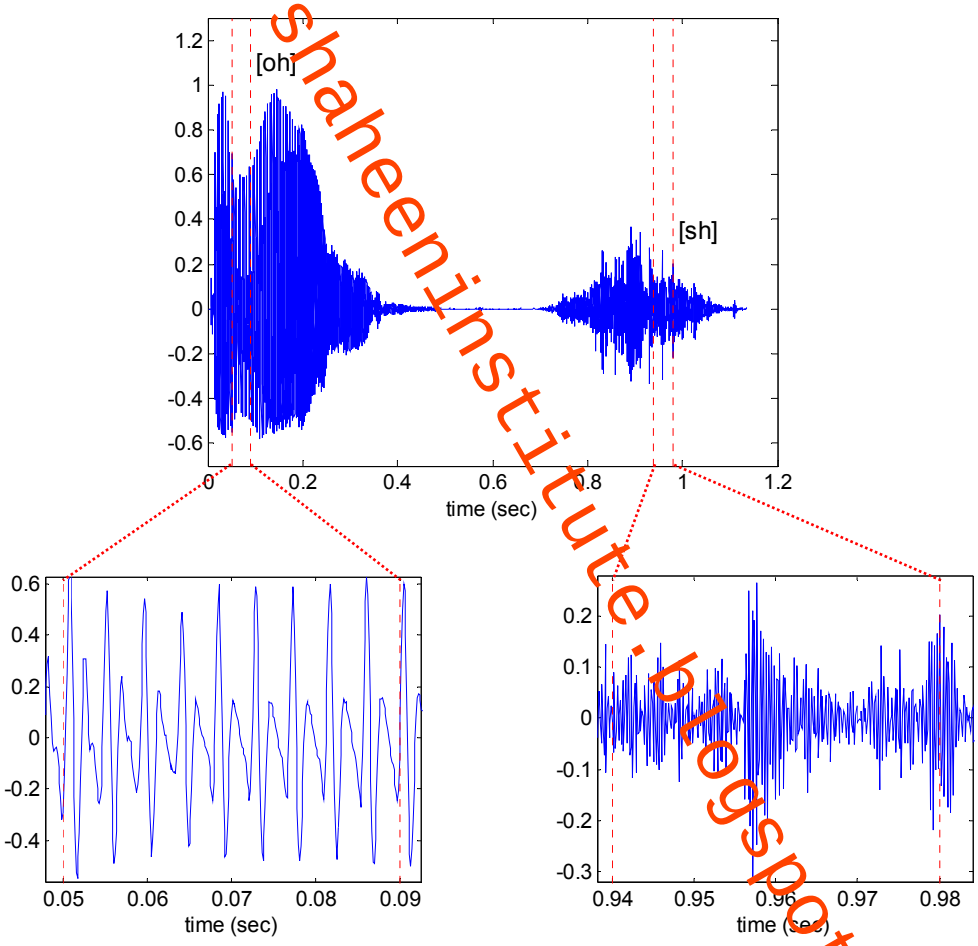

Figure 2. Example of voiced [oh] and unvoiced [sh] sounds spoken by a female speaker

Since voiced sounds are periodic and unvoiced sounds are not, measuring the periodic similarity between samples in consecutive pitch cycles can give a reasonable indication of the voicing of the signal. The Pitch Similarity measurement (P_S) can be computed by [4]

$$P_S = \frac{\left[\sum_{n=0}^{N-1} s(n)s(n-T)\right]^2}{\sum_{n=0}^{N-1} s^2(n)\ \sum_{n=0}^{N-1} s^2(n-T)} \tag{2.1}$$

where T is the pitch period and N is the number of samples per frame. Pitch period estimation is presented in Sub-Section 3.1.1 of the next chapter.

P_S values vary between 0 and 1, indicating no similarity and 100% similarity respectively. Figure 3 shows a time plot of the waveform of the word [psychology] against the pitch similarity. The plot shows that the voiced parts of the speech have higher pitch similarity than the unvoiced parts.

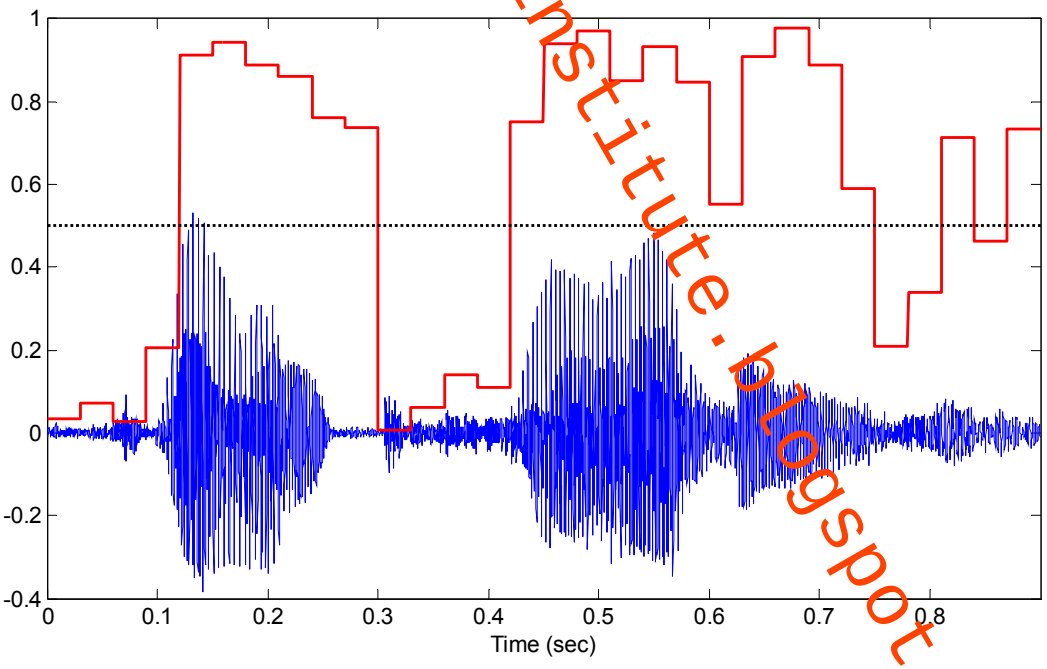

Figure 3. **Speech waveform and the corresponding pitch similarity plot with a possible voicing threshold of 0.5 (shown by the dashed line)**

2.2.2. Short time energy

Generally, the amplitude of unvoiced speech segments is much lower than the amplitude of voiced segments, (e.g. see Figure 2). The energy of the speech signal provides a representation that reflects these amplitude variations. The short-time energy e of an N–sample frame is defined as:

$$e = \sum_{n=0}^{N-1} s^2(n) \tag{2.2}$$

where $s(n), n = 0, 1, \ldots, N-1$ is one speech frame.

Typically, voiced sounds have higher energy than unvoiced ones [3].

It can be seen in Figure 4 that the short time energy of the voiced parts of the word [psychology] is higher than the energy for the unvoiced parts.

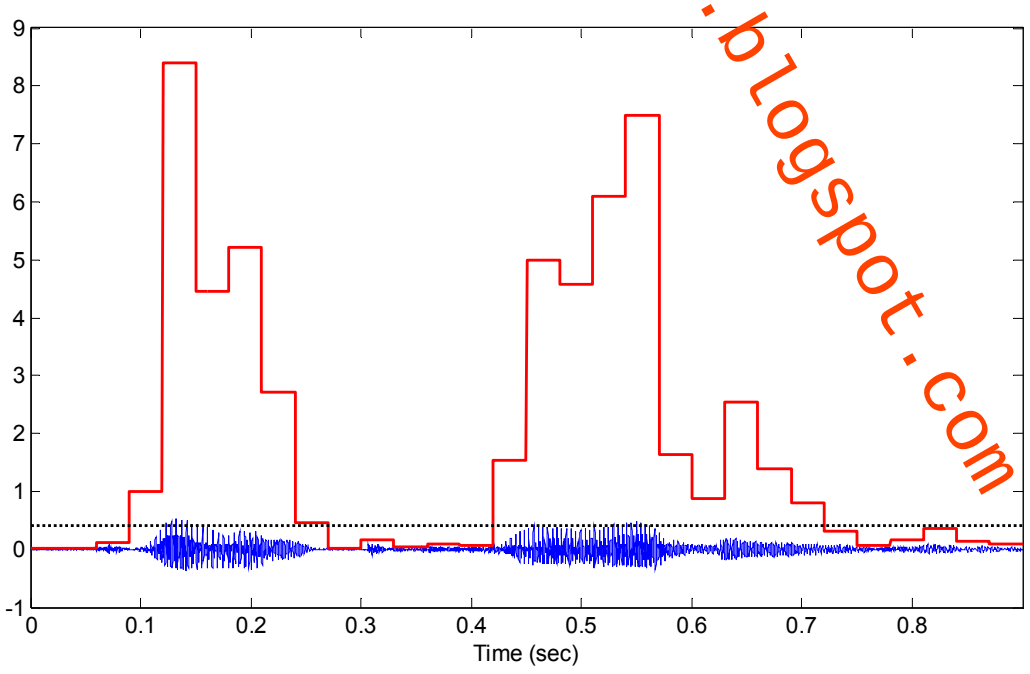

Figure 4. **Speech waveform and the corresponding short time energy plot with a possible voicing threshold of 0.4 (shown by the dashed line)**

2.2.3. Zero crossing rate

In the context of discrete-time signals, a zero crossing is said to occur if successive samples have different algebraic signs. The Zero Crossing Rate (ZCR) is the number of times in a given time interval/frame that the amplitude of the speech signal crosses zero.

$$ZCR = \frac{1}{2N}\sum_{n=0}^{N-1}|sgn[s(n)] - sgn[s(n+1)]| \tag{2.3}$$

where

$$sgn[s(n)] = \begin{cases} 1 & s(n) \geq 0 \\ -1 & s(n) < 0 \end{cases} \tag{2.4}$$

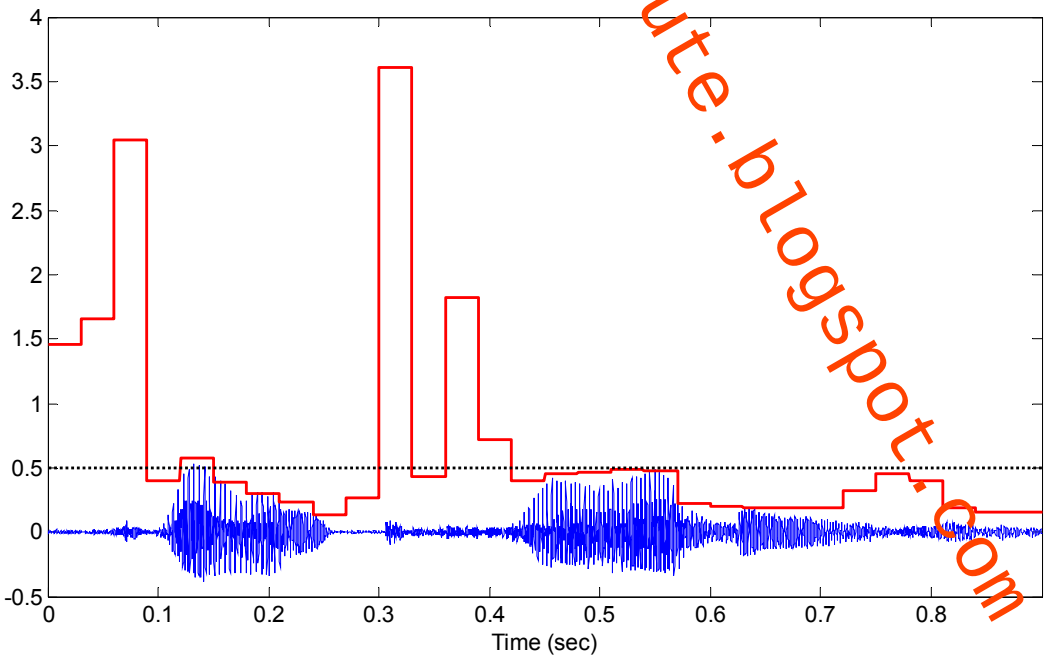

Figure 5. **Speech waveform and the corresponding zero crossing rate plot with a possible voicing threshold of 0.5 (shown by the dashed line)**

Unvoiced speech has random characteristics causing it to oscillate much faster than voiced speech [3]. ZCR also depend on the signal pitch (for voiced sounds); e.g., ZCR for voiced female speech is higher than that for voiced male speech [4], which can result in a bias voicing decision for voiced female speech. Therefore a simple pitch weighting can be used to weight the decision threshold [4]. Figure 5 above shows an example of the ZCR criterion for the word [psychology] by a female speaker; the ZCR is weighted by multiplying it by the pitch period of the frame to enhance the decision threshold.

2.2.4. Spectrum tilt

Voiced speech has higher energy in low frequencies and unvoiced speech usually has higher energy in high frequencies resulting in opposite spectral tilts; the spectral tilts can be represented by the first order normalized autocorrelation coefficient [4].

The Spectral tilt (S_t) can be calculated by

$$S_t = \frac{\sum_{n=0}^{N-1} s(n)s(n-1)}{\sum_{n=0}^{N-1} s^2(n)} \tag{2.5}$$

Figure 6 shows the classification of a speech segment using the spectral tilt criterion.

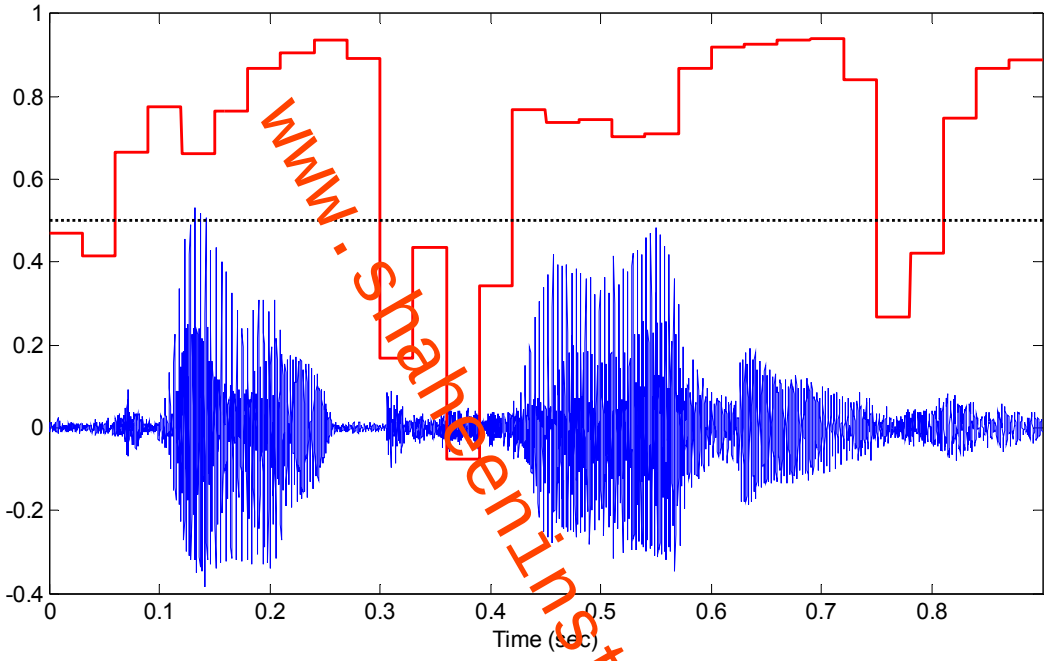

Figure 6. **Speech waveform and the corresponding spectrum tilt plot with a possible voicing threshold of 0.5 (shown by the dashed line)**

Decision making

The above decision criteria along with other criteria [4] are used to take the frame's voicing decision. Sometimes it is not absolutely clear if a frame is voiced or unvoiced especially for transitional frames (frames during the transition from voiced to unvoiced sounds and vice versa) making it difficult to judge the frame as strictly voiced or strictly unvoiced. The simplest decision making rule would be to use a majority vote [4], that is to use many decision criteria then make a combined decision. Some frames are harder to classify than others, however, it is still important to classify the frames as accurately as possible in order to correctly reproduce a high quality speech as will be described in the next chapter.

3.0 SPEECH CODING

Speech coding, or speech compression, plays an important role in modern voice-enabled technologies like digital speech communication, voice over Internet protocol and voice storage. Speech coding is the process where a raw speech signal is digitally represented with as few bits as possible while preserving a reasonable level of quality for the reconstructed (synthesized) speech [1]. Speech coding systems attempt to achieve a compromise between compression, quality and complexity.

Traditionally, most speech coding systems are designed to support telecommunication applications with frequency limited between 300 and 3400 Hz [1]. Since the sampling frequency is at least twice the bandwidth of the signal, according to Nyquist theorem, a sampling frequency of 8 kHz is commonly used as a standard sampling frequency for speech signals.

Speech coding techniques can be broadly divided into two classes, waveform and parametric coding methods [4]. Waveform coders attempt to produce a reconstructed signal whose waveform is as close as possible to the original speech waveform. Parametric coders, also known as vocoders, try to extract the parameters of the model that is responsible for generating the speech signal. Waveform coders are able to produce high quality speech at high bit rates; vocoders however are able to generate intelligible, yet not so natural sounding, speech at much lower bit rates.

This chapter is devoted to studying vocoders that are based on a linear prediction model. The linear prediction problem is introduced in Section 3.1 and the autocorrelation solution to the problem is studied. Linear prediction vocoders are also presented. Those coders basically receive a raw sampled speech signal and analyze it in a frame by frame manner. The output parameters

of linear prediction coders are the voiced/unvoiced decision, the all-pole filter coefficients, the pitch period and the gain. These parameters are then quantized and sent over the transmission channel to be used at the receiver to generate a synthetic version of the input speech. Although the linear prediction model is very basic and results in a low bit rate, below 2.5 kbits/sec, the resultant synthesized speech is not of a high quality, does not sound natural and suffers annoying artifacts such as buzzes, cracks and tonal noises because of the degradation due to errors in pitch estimation and voiced/unvoiced decisions [1].

In order to improve the quality of the synthesized speech, a multi-pulse excitation model [5], described in Section 3.2, suggests quantizing and sending the linear prediction filter coefficients along with a multi-pulse excitation sequence. The coefficients and the excitations are then used at the receiver end to synthesize the speech. This approach increases the quality of the synthesized speech with bit rates below 16 kbits/sec.

Section 3.3 introduces robust linear prediction where different methods of finding better linear prediction coefficients are presented.

3.1 LINEAR PREDICTION CODING

Linear Prediction (LP) methods can be viewed as redundancy removal procedures where repeated/predictable information in a signal is eliminated. Redundancy elimination results in signal compression since the number of bits required to represent the information is reduced [1].

Linear prediction is one of the most useful linear prediction based speech analysis models. It is widely used for encoding speech at low bit rates and yet provides very accurate estimates of the speech parameters [3]. LP based vocoders are designed to simulate the human

speech production mechanism [4], where the vocal tract is modeled by a linear prediction filter $H(z)$ as shown in Figure 7. $H(z)$ is excited by either a quasi-periodic pulse train with impulses located at pitch period intervals for voiced speech production, or by random noise for unvoiced speech production.

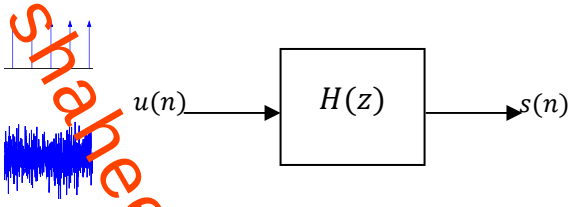

Figure 7. Discrete speech production model [6]

The basic idea behind LP analysis is that a speech signal $s(n)$ can be approximated by a linear combination of past samples of the signal $s(n)$ and past and present samples of an unknown input $u(n)$ such that:

$$s(n) = -\sum_{k=1}^{p} a_k s(n-k) + G\sum_{l=0}^{q} b_l u(n-l) \ , \qquad b_0 = 1 \tag{3.1}$$

where $a_k, 1 \le k \le p$, b_l , $1 \le l \le q$ and the gain G are the parameters of the hypothesized system [6].

In the frequency domain, equation (3.1) becomes:

$$H(z) = \frac{S(z)}{U(z)} = G\frac{1 + \sum_{l=1}^{q} b_l z^{-l}}{1 + \sum_{k=1}^{p} a_k z^{-k}} \tag{3.2}$$

$$S(z) = \sum_{n=-\infty}^{\infty} s(n) z^{-n} \tag{3.3}$$

where $S(z)$ is the z transform of $s(n)$, $U(z)$ is the z transform of $u(n)$, and $H(z)$ is the same transfer function of the system in Figure 7. $H(z)$ in equation (3.2) is a general pole-zero model which has two interesting special cases:

- The all-zero, moving average (MA), model: $a_k = 0$ for $1 \leq k \leq p$
- The all-pole, autoregressive (AR), model: $b_l = 0$ for $1 \leq l \leq q$

Autoregressive models are known to well represent voiced speech signals while pole-zero models are needed for unvoiced speech signals [2]. However, when the prediction order p is high enough, all-pole models effectively represent all types of speech signals [3]; thus we only examine all-pole models where the speech signal is a linear combination of its past values and some input $u(n)$

$$s(n) = -\sum_{k=1}^{p} a_k s(n-k) + Gu(n) \tag{3.4}$$

Hence $H(z)$ is defined as:

$$H(z) = \frac{S(z)}{U(z)} = \frac{G}{1 + \sum_{k=1}^{p} a_k z^{-k}} = \frac{G}{A(z)}$$
$$A(z) = 1 + \sum_{k=1}^{p} a_k z^{-k} \tag{3.5}$$

3.1.1 The Liner Prediction Problem

Linear prediction can be described as a system identification problem, where the parameters of an AR model are estimated from the signal itself [4]. A simple block diagram of the linear predictive model of the speech signal is shown in Figure 8; where the AR filter is excited by the output of a voiced/unvoiced switch.

From equation (3.4) and assuming that the input $u(n)$ is totally unknown, the problem of linear prediction is to predict the AR parameters, also known as the Linear Prediction Coefficients (LPCs), a_k, the gain G and the pitch period that corresponds to the speech production model that best approximates the signal $s(n)$ from its past samples.

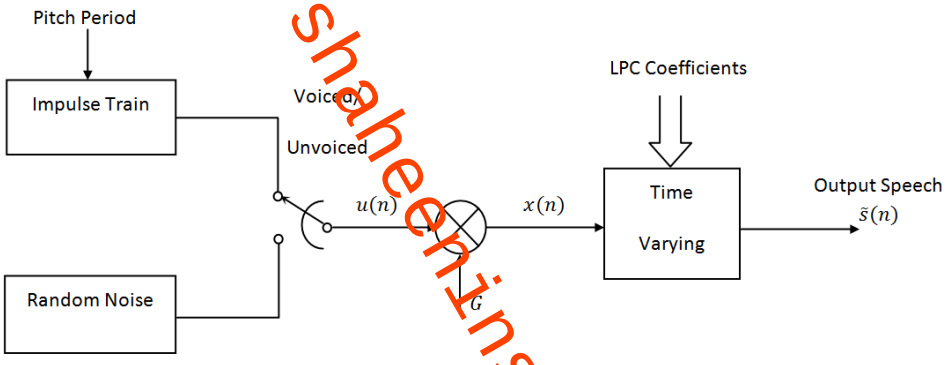

Figure 8. Block diagram of the simplified LPC speech production model [4]

The approximated signal $\tilde{s}(n)$ is thus defined as:

$$\tilde{s}(n) = -\sum_{k=1}^{p} a_k s(n-k) \tag{3.6}$$

Then the prediction error, referred to as the residual, is:

$$e(n) = s(n) - \tilde{s}(n) = s(n) + \sum_{k=1}^{p} a_k\, s(n-k) \tag{3.7}$$

Using the method of least squares, the LPCs are found by minimizing the mean squared error,

$$E = \mathcal{E}\{e^2(n)\} = \mathcal{E}\left\{s(n) + \sum_{k=1}^{p} a_k\, s(n-k)\right\}^2 \tag{3.8}$$

E is minimized by setting its partial derivatives with respect to a_k to zero,

$$\frac{\partial E}{\partial a_i} = 2\mathcal{E}\left\{\left(s(n) + \sum_{k=1}^{p} a_k\, s(n-k)\right) s(n-i)\right\} = 0\,, \qquad 1 \le i \le p \tag{3.9}$$

Rearranging (3.9), we get:

$$\mathcal{E}\{s(n)s(n-i)\} + \mathcal{E}\left\{\sum_{k=1}^{p} a_k\, s(n-k)s(n-i)\right\} = 0$$

$$\mathcal{E}\{s(n)s(n-i)\} + \sum_{k=1}^{p} a_k\, \mathcal{E}\{s(n-k)s(n-i)\} = 0 \tag{3.10}$$

Equation (3.10) can be written in terms of the autocorrelation and is known as the LPC analysis equation:

$$\sum_{k=1}^{p} a_k R(i-k) = -R(i), \qquad 1 \le i \le p \tag{3.11}$$

where $R(i)$ is the autocorrelation of the signal $s(n)$ and

$$R(i-k) = \mathcal{E}\{s(n-k)s(n-i)\} \tag{3.12}$$

Expanding (3.8) and substituting (3.10), the minimum average error is given by

$$E = \mathcal{E}\{s^2(n)\} + \sum_{k=1}^{p} a_k\, \mathcal{E}\{s(n)s(n-k)\} \tag{3.13}$$

This derivation is valid for stationary signals, deterministic or random; however, the speech signal has a dynamic nature making its characteristics vary with time. Therefore, LPC analysis must be performed on blocks or frames of speech where the signal's statistical properties are almost unchanged. Thus the LPCs are calculated for every signal frame using the above procedure since the signal is believed to be locally stationary within that frame. To emphasize that the analysis is performed every frame m of the signal $s(n)$, a subscript, m, will be added to the signal, residual and autocorrelation expressions. Rewriting the predicted signal, the prediction error and the LPC analysis equations:

$$\tilde{s}_m(n) = -\sum_{k=1}^{p} a_k s_m(n-k) \tag{3.14}$$

$$e_m(n) = s_m(n) - \tilde{s}_m(n) = s_m(n) + \sum_{k=1}^{p} a_k\, s_m(n-k) \tag{3.15}$$

$$\mathcal{E}\{s_m(n)s_m(n-i)\} + \sum_{k=1}^{p} a_k\, \mathcal{E}\{s_m(n-k)s_m(n-i)\} = 0 \tag{3.16}$$

where, $$R_m(i-k) = \mathcal{E}\{s_m(n-k)s_m(n-i)\} \tag{3.17}$$

$\therefore$ $$\sum_{k=1}^{p} a_k R_m(i-k) = -R_m(i), \qquad 1 \le i \le p \tag{3.18}$$

$$E_m = R_m(0) + \sum_{k=1}^{p} a_k\, R_m(k) \tag{3.19}$$

where m is a frame of N samples. Typically the frame length N is of 16 to 32 ms of speech [4], which is 128 to 256 samples at a sampling frequency of 8 kHz. A longer frame has the advantage of less computational complexity and lower bit-rate, since the calculation and transmission of LPCs are done less frequently. However due to the changing nature of speech, the LPCs derived from longer frames might not be able to produce good approximation of the speech.

3.1.1.a. Linear prediction coefficients (Autocorrelation method)

Linear prediction coefficients can be solved for using several methods; one of which is the autocorrelation method [3]. The main advantage of this method is its stability [6], where all the roots of the polynomial $A(z)$ fall inside the unit circle and thus the system $H(z)$ in Equation (3.5) is guaranteed to remain stable. The method's name comes from the autocorrelation term in Equation (3.18), which can be written in a matrix form as:

$$\mathbf{R_m}\boldsymbol{a} = -\boldsymbol{r_m} \tag{3.20}$$

equivalently,

$$\begin{bmatrix} R_m(0) & R_m(1) & \cdots & R_m(p-1) \\ R_m(1) & R_m(0) & \cdots & R_m(p-2) \\ \vdots & & \ddots & \vdots \\ R_m(p-1) & R_m(p-2) & \cdots & R_m(0) \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_p \end{bmatrix} = -\begin{bmatrix} R_m(1) \\ R_m(2) \\ \vdots \\ R_m(p) \end{bmatrix} \tag{3.21}$$

Equation (3.20) can be solved for the LPCs, $\boldsymbol{a}$, by finding the matrix inverse of $\mathbf{R_m}$; unfortunately, matrix inversion is generally expensive in terms of computation specially for higher p. However, efficient and neat recursive algorithms have been developed to solve (3.20) taking advantage of its elegant structure. Durbin's recursive procedure is believed to be one of the most efficient algorithms to solve the LPC analysis equation [3].

Durbin's recursive Algorithm [6]:

- Initialize: $E_m^{(0)} = R_m(0)$
- *for* $i = 1, 2, \ldots, p$

$$\varkappa_i = -\left[R_m(i) + \sum_{j=1}^{i-1} a_j^{(i-1)} R_m(i-j)\right] / E_m^{(i-1)}$$

$$a_i^{(i)} = \varkappa_i$$

$$a_j^{(i)} = a_j^{(i-1)} + \varkappa_i a_{i-j}^{(i-1)}, \quad 1 \le j \le i-1$$

$$E_m^{(i)} = (1 - \varkappa_i^2) E_m^{(i-1)}$$

end

- Final solution: $a_j = a_j^{(p)}, \ 1 \le j \le p$

where $\varkappa_i$'s are known as the reflection coefficients

It can be noted that in obtaining the solution for a predictor of order p, one actually computes all the predictors of order less than p. Furthermore; at each step i, the minimum total error $E_m^{(i)}$, is calculated and thus can be monitored as the predictor order is increased.

3.1.1.b. Computation of the gain

The speech production model in Figure 8 shows the model gain as a scalar factor that is multiplied by the input $u(n)$ to assign the frame energy. Equation (3.4) relates the gain factor to the LPCs as:

$$Gu(n) = s(n) + \sum_{k=1}^{p} a_k s(n-k) \tag{3.22}$$

where $u(n)$ is either a unit impulse $\delta(n)$ for voiced speech or a zero mean unit variance white noise for unvoiced speech. The gain G is therefore derived for the voiced/unvoiced cases.

For voiced speech, $u(n) = \delta(n)$ and equation (3.22) can be written as

$$G\delta(n) = s(n) + \sum_{k=1}^{p} a_k s(n-k) \tag{3.23}$$

Multiplying (3.23) by $s(n)$ and summing over n:

$$G \sum_{n} s(n)\, \delta(n) = \sum_{n} s^2(n) + \sum_{k=1}^{p} a_k \sum_{n} s(n)\, s(n-k) \tag{3.24}$$

at $n = 0$, from (3.23), $s(0) = G$ and thus the left hand side of (3.24) is G^2

$$G^2 = R(0) + \sum_{k=1}^{p} a_k R(k) \tag{3.25}$$

For unvoiced speech, $u(n)$ is white noise with $\mathcal{E}\{u(n)\} = 0$, $\mathcal{E}\{u^2(n)\} = 1$. Writing the autocorrelation function for the speech signal:

$$R(i) = \mathcal{E}\{s(n)s(n-i)\}$$

$$R(i) = G\mathcal{E}\{u(n)s(n-i)\} - \sum_{k=1}^{p} a_k \mathcal{E}\{s(n-i)s(n-k)\} \tag{3.26}$$

At $i = 0$,

$$R(0) = G\mathcal{E}\{u(n)s(n)\} - \sum_{k=1}^{p} a_k \mathcal{E}\{s(n)s(n-k)\} \tag{3.27}$$

$$R(0) = G\mathcal{E}\left\{u(n)\left[Gu(n) - \sum_{k=1}^{p} a_k s(n-k)\right]\right\} - \sum_{k=1}^{p} a_k R(k) \tag{3.28}$$

$$R(0) = G^2\mathcal{E}\{u^2(n)\} - G\sum_{k=1}^{p} a_k \mathcal{E}\{u(n)s(n-k)\} - \sum_{k=1}^{p} a_k R(k) \tag{3.29}$$

Since $u(n)$ is independent of $s(n-k)$

$$R(0) = G^2\mathcal{E}\{u^2(n)\} - G\sum_{k=1}^{p} a_k \mathcal{E}\{u(n)\}\mathcal{E}\{s(n-k)\} - \sum_{k=1}^{p} a_k R(k) \tag{3.30}$$

Hence,

$$\begin{aligned} R(0) &= G^2 - \sum_{k=1}^{p} a_k R(k) \\ G^2 &= R(0) + \sum_{k=1}^{p} a_k R(k) \end{aligned} \tag{3.31}$$

Which is the same result obtained for the voiced speech case in equation (3.25).

3.1.1.c. Pitch period estimation

In the case of voiced speech frames, time length between consecutive excitation impulses is known as the fundamental period or the pitch period. For men, the possible pitch frequency range is usually between 50 and 250 Hz, while for women it is between 120 and 500 Hz [1].

Pitch period estimation is essential for LP coding because the periodic excitation for voiced sounds is generated by switching an electric switch on every pitch period. Hence it is important to accurately estimate the pitch period in order to synthesize a high quality speech.

There are several ways to estimate the pitch period of a frame; one of the most common methods uses the autocorrelation function [1]. The autocorrelation function $R(l,m)$ is calculated for the speech frame $s(n)$ of length N that ends at the time instant m.

$$R(l,m) = \sum_{n=m-N+1}^{m} s(n)s(n-l) \tag{3.32}$$

where l is the time lag.

The autocorrelation is calculated over the entire range of lag, from 0 to N, and the pitch period is the lag that corresponds to the highest autocorrelation.

Another way that is more preferable since it doesn't require multiplications that are considered computationally expensive uses the Magnitude Difference Function (MDF) which is calculated using a similar formulation as (3.32) but with a subtraction instead of a multiplication.

$$MDF(l,m) = \sum_{n=m-N+1}^{m} |s(n) - s(n-l)| \tag{3.33}$$

The pitch period in this case is the time lag that corresponds to the lowest MDF.

3.1.2 The Linear Prediction Coefficient Vocoder

Once the linear prediction problem is solved and all the LP coding parameters (voicing decision, pitch period, model gain and LPCs) are found, the model shown in Figure. 8 is fully defined and the parameters are ready to be properly quantized and sent over the transmission channel.

The voicing parameter, pitch period and the gain are directly quantized, coded and sent over the channel. 1 bit is enough to quantize the voiced/unvoiced parameter, 6 bits are sufficient to quantize the pitch period, and about 5 bits are required for quantizing the gain [3].

However, the LPCs are very sensitive to quantization; small changes made to the LPCs may result in the filter being unstable, which means more bits are needed to adequately quantize them. It was found that almost 8-10 bits per coefficient are required to quantize the LPCs with an accepted accuracy [3] which is not efficient for low bit rates. Therefore LPCs are not quantized directly, instead a proper representation that is less sensitive to small changes is quantized. Representations such as line spectral frequency (LSF), the predictor polynomial roots and the reflection coefficients had been introduced and used for LPC quantization coding the LPCs with about 40-50 bits [7].

For a frame of about 30 ms almost 65 bits are required to code all the LPC model parameters resulting in a total bit rate of about 2.2 kbits/sec [3]. The LPC model has a relatively low computational cost and results in a low bit-rate speech coding. However, the LPC model is also highly inaccurate in various circumstances resulting in a low quality synthetic speech.

One of major limitation of the LPC model is due to the misclassification of speech frames into strictly voiced or unvoiced, as discussed in Section 2 of the previous chapter. Misclassifying the speech frames results in an incorrect modeling of the LP filter excitations by strictly random noise or strictly periodic impulse train. This inaccuracy in the voicing decision thus results in annoying artifacts such as buzzes and tonal noises in the synthetic speech [1].

3.2 MULTI-PULSE EXCITED LINEAR PREDICTION CODING

Multi-pulse excited linear prediction coding (MPLPC) was first introduced by Atal and Remede [5] as a new speech production model that generates natural sounding speech at a low bit rate. As the name implies, the excitation signal of the MPLPC consists of a sequence of pulses whose amplitudes and positions are selected to minimize an error criterion with no preference or a priori knowledge of the voicing nature of the speech segment.

Figure 9 shows a block diagram of the MPLPC. The diagram is similar to the conventional LPC one; the only difference is the absence of the voiced/unvoiced switch and the quasi periodic/white noise generators which are replaced by a multi-pulse excitation generator.

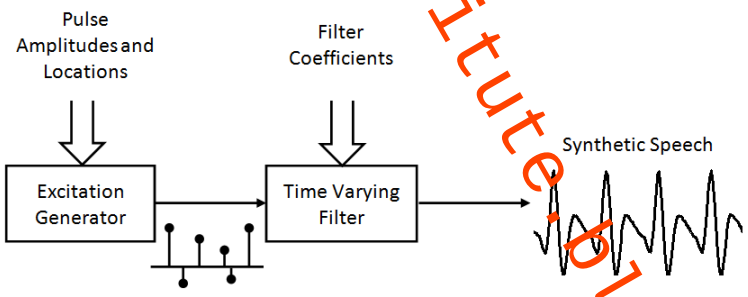

Figure 9. Block diagram of a MPLPC speech synthesis model [8]

The excitation signal is a sequence of pulses located at times $m_1, m_2, \ldots, m_K$ with amplitudes $g_1, g_2, \ldots, g_K$. The K pulse amplitudes and locations are sent every frame over the transmission channel along with the filter coefficients. The multi-pulse signal is then used to excite a synthesis filter to reproduce the speech signal.

The time varying filter is typically a linear prediction all-pole filter whose coefficients are obtained as described in Section 3.1. The pulse amplitudes and locations are found by an analysis-by-synthesis procedure [5] shown in the block diagram of Figure 10 where a multi-pulse

signal is used to excite an LPC filter which generates a synthesized speech; the synthetic speech is compared to the original speech to produce an error signal which is then properly weighted and used as an error criterion. The pulse locations and amplitudes are found so they minimize the mean squared weighted error.

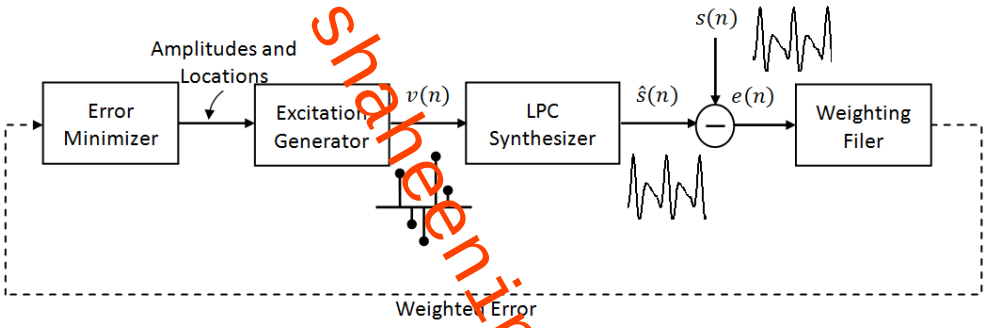

Figure 10. Analysis by synthesis block diagram for finding amplitudes and locations of multi-pulse excitation [5]

Atala and Remedy [5] suggested that since energy is highly concentrated in the formant regions, one can tolerate more error in those regions than in regions in between them; therefore a weighting filter is placed to de-emphasize the error in the formant regions. The frequency characteristics, in the z-transform, of the weighting filter is given by:

$$W(z) = \frac{1 + \sum_{i=1}^{P} a_i z^{-i}}{1 + \sum_{i=1}^{P} a_i r^i z^{-i}} \qquad (3.34)$$

where a_i's are the LPCs and r is a fraction between 0 and 1 that controls the error increase in the formant regions. The value of r is determined by the degree to which one wishes to de-emphasize the noise in the formant regions; setting r to 0.8 at a sampling rate of 8 kHz is proved to be suitable [5].

3.2.1 Pulse Search Procedure

The amplitudes and locations of the excitation signal are found such that they minimize the mean squared weighted error. The synthesized signal is expressed in terms of the multi-pulse excitation sequence of amplitudes g_i and locations m_i as

$$\hat{s}(n) = \sum_{i=1}^{K} g_i \,.\, h(n - m_i) \tag{3.35}$$

where $h(n)$ is the impulse response of the LPC filter.

Using a weighting filter $W(z)$ with an impulse response $w(n)$, the total weighted squared error between the original and synthesized speech is

$$E = \sum_{n=0}^{N-1} \left[s_w(n) - \sum_{i=1}^{K} g_i \,.\, h_w(n - m_i) \right]^2 \tag{3.36}$$

where

$$\begin{aligned} s_w(n) &= s(n) * w(n) \\ h_w(n) &= h(n) * w(n) \end{aligned} \tag{3.37}$$

Finding all the amplitudes and locations at once is extremely complex therefore a sub-optimal procedure was proposed [5] where the pulses are searched for one pulse at a time over a short time segment, typically 5 to 10 ms, where when searching for the pulse k, one assumes that all the previous $k-1$ pulses amplitudes and locations are known.

Minimizing (3.36) with respect to g_k(setting the derivative to zero), g_k is found to be:

$$g_k = \frac{R_{hx}(m_k) - \sum_{i=1}^{k-1} g_i R_{hh}(m_k, m_i)}{R_{hh}(m_k, m_k)} \quad , \quad \begin{matrix} 0 \le m_i \\ m_k \le N-1 \end{matrix} \tag{3.38}$$

where,

$$R_{hx}(m_k) = \sum_{n=0}^{N-1} s_w(n).h_w(n-m_k) \quad , \qquad 0 \le m_k \le N-1 \tag{3.39}$$

$$R_{hh}(m_k, m_i) = \sum_{n=0}^{N-1} h_w(n-m_i).h_w(n-m_k) \quad , \qquad \begin{matrix} 0 \le m_k \\ m_i \le N-1 \end{matrix} \tag{3.40}$$

Pulse Search Algorithm [8]:

- Initialize: $R(m) = R_{hx}(m)$ for $0 \le m \le N-1$
- Search: *for* $k = 1 : K$
  - Find the pulse location m_k that maximizes $|R(m)|$
  - Find the pulse amplitude g_k using (3.38)
  - Update: $R(m) = R(m) - g_k R_{hh}(m - m_k)$

end

This is a basic pulse search process, where the pulse that minimizes the total error is searched for, then its contribution to the error is subtracted and the next pulse is searched for.

3.2.2 Improved (Amplitude Updating) Pulse Search Method

It was observed that finding the amplitudes and locations of the pulses in a successive manner is inaccurate for closely spaced pulses; however avoiding this inaccuracy is possible by updating all the amplitudes after obtaining the positions so that the updated amplitudes minimize the error criterion [9].

Finding the derivative of (3.36) with respect to g_j and setting it to zero

$$\sum_{j=1}^{K} g_j R_{hh}(m_i, m_j) = R_{hx}(m_i)\,, \qquad \text{for} \qquad 1 \le i \le K \tag{3.41}$$

Given that all the pulses locations are now known, the updated amplitudes are found by solving (3.41) for g_j's.

The MPLPC model is shown to produce high quality natural sounding speech at medium bit rate, 10 to 16 kbits/sec [8]. Figure 11 below shows the effective performance of the MPLPC; a speech signal is well modeled by the multi-pulse excitation signal resulting in a speech waveform that well approximates the original signal especially the pitch characteristics.

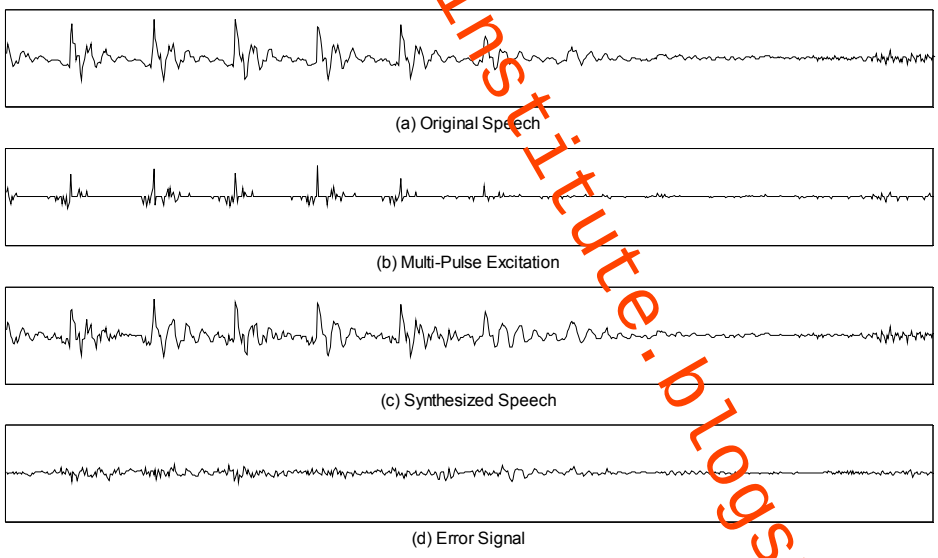

Figure 11. Waveform illustration of the MPLPC coder

3.3 ROBUST LINEAR PREDICTION CODING

As described in Section 3.1 LP finds the inverse filter coefficients a_k's such that $A(z) = 1 + + \sum_{k=1}^{p} a_k z^{-k}$. Passing the speech signal through $A(z)$ results in the residual signal $e(n)$, which represents the pitch information in the speech. On the other hand, the magnitude spectrum of

$1/A(z)$ describes the spectral envelope of the speech signal thus contains formant information [10]. This is illustrated in Figure 12 which shows a residual signal of a voiced speech segment (a) and the spectrum of the same speech segment and the LP filter (b).

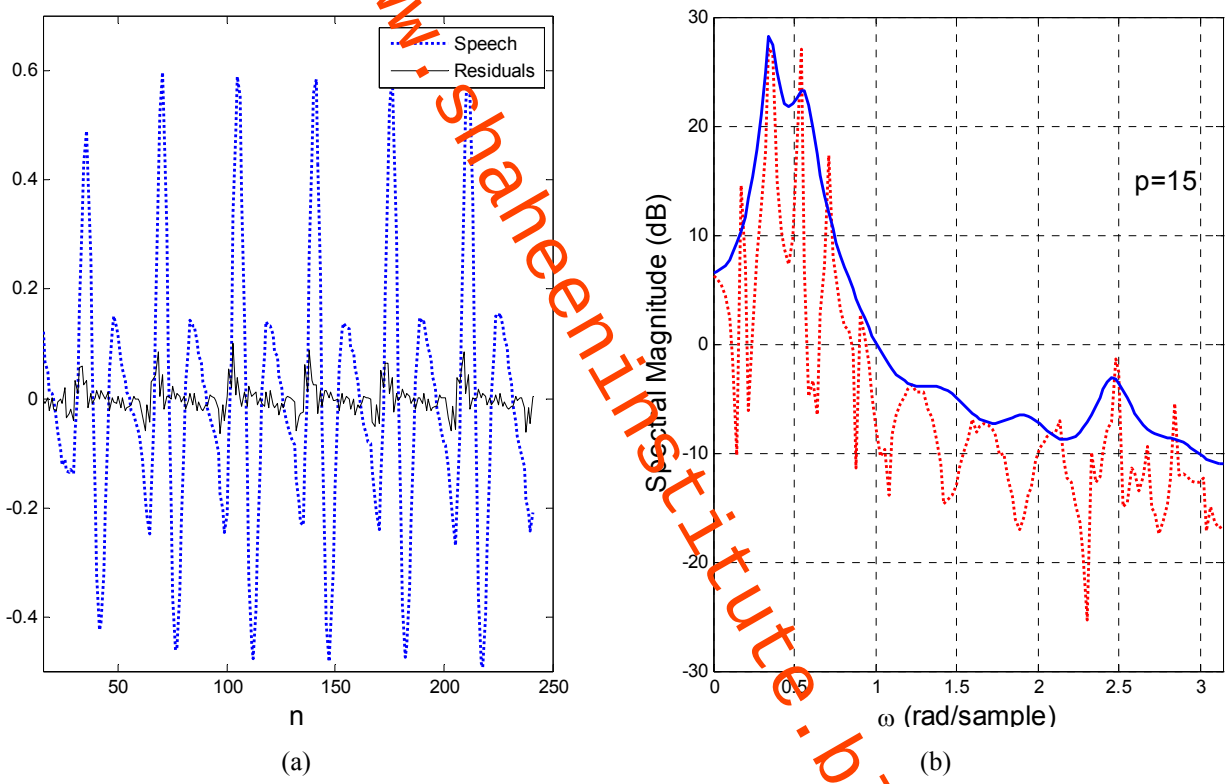

Figure 12. **Pitch and vocal tract information captured by LP analysis**
(a) Pitch information in the residual signal (b) Formant information in the filter coefficients

The success of LP methods depends on determining the coefficients a_k's such that $A(z)$ best captures the vocal tract information and the LP residual contains the pitch information. Further, LP methods must be robust to noise so that the vocal tract information is well extracted even for noisy speech. It has been observed that the conventional method of LP analysis based on squared error is sensitive to noisy speech [11].

The Robust Linear Prediction (RBLP) procedure takes into account the non-Gaussian nature of the source excitation for voiced speech by assuming that the innovation is from a mixture distribution, such that a large portion of the excitations is from a normal distribution

with a small variance while a small portion of the glottal excitations is from an unknown distribution with a much bigger variance [12].

The RBLP procedure minimizes the sum of weighted residuals, rather than minimizing the sum of squared residuals. The assigned weight is a function of the prediction residual and the cost function can be selected to assign more weight to the bulk of small residuals while down-weighting the small portion of large residuals.

A robust estimate of the LP coefficients is hence obtained by solving the following optimization problem [12]:

$$\hat{\boldsymbol{a}} = \min_{\boldsymbol{a}} \sum_{n=p+1}^{N} \rho(e_n(\boldsymbol{a})) \tag{3.42}$$

where,
$$e_n(\boldsymbol{a}) = s_n + \sum_{k=1}^{p} a_k\, s_{n-k} \qquad n = p+1\,, \dots\,, N \tag{3.43}$$

$\rho(x)$ is an appropriate loss function that has a bounded derivative, psi-function, $\psi(x) = \rho'(x)$.

Huber's psi- function, $\psi_H(x)$, is used to find the minimum due to its robustness properties since the function is bounded monotonically non-decreasing, which yields uniqueness [12]. The effect of using $\psi_H(x)$ is to assign less weight to the small portion of large residuals so that the outliers will not terribly influence the final estimate, while giving unity weight to the bulk of small to moderate residuals. Huber's psi is defined as:

$$\psi_H(x) = \min[c\,, \max(x\,, -c)] \tag{3.44}$$

where c is an efficiency tuning constant.

The associated Huber's loss function is thus defines as:

$$\rho_H(x) = \begin{cases} x^2/2 & \text{if } |x| \le c \\ c|x| - c^2/2 & \text{if } |x| > c \end{cases} \tag{3.45}$$

In other words, Huber's loss is a quadratic function in the middle and an absolute value function at the tails, which results in more minimization of the small errors while allowing large errors to grow larger. Setting the derivative of (3.42) to zeros,

$$\sum_{n=p+1}^{N} s_{n-j}\psi(e_n(\boldsymbol{a})) = 0 \qquad j = 1, 2, \dots, p \tag{3.46}$$

The LP coefficients $\boldsymbol{a}$ are found by solving the set of non-linear equations (3.46); the following sub-sections discuss two different approaches for the solution.

3.3.1 Solving the RBLP problem by Iterative Reweighted Least Squares Algorithm [12]

The system of non-linear equations in (3.46) requires iterative methods to solve for the coefficients. Given a preliminary estimate $\hat{\boldsymbol{a}}$ (usually the conventional LPC).

Often, $\psi'(x)$ is approximated by a weight function $W(x)$, where

$$W(x) = \psi(x)/x \tag{3.47}$$

Weighting the residuals by $W(x)$ in the estimating equation, Equation (3.46), we get,

$$\sum_{n=p+1}^{N} s_{n-j}\, e_n\left(\boldsymbol{a}^{(k+1)}\right) W\left(e_n(\boldsymbol{a}^{(k)})\right) = 0 \qquad 1 \le j \le p \tag{3.48}$$

where k is the iteration number and $e_n(\boldsymbol{a})$ is residuals defined as in (3.43).

Defining C^{**} and c^{**} as:

$$\begin{aligned} C_{ij}^{**} &= \sum_{n=p+1}^{N} s_{n-j}\, s_{n-i}\, W\left(e_n(\boldsymbol{a}^{(k)})\right) \qquad 1 \le i, j \le p \\ c_j^{**} &= \sum_{n=p+1}^{N} s_{n-j}\, s_n\, W\left(e_n(\boldsymbol{a}^{(k)})\right) \qquad 1 \le j \le p \end{aligned} \tag{3.49}$$

(3.48) can be written in a matrix form as:

$$\boldsymbol{C}^{**}\,\boldsymbol{a}^{(\boldsymbol{k+1})} = -\boldsymbol{c}^{**} \tag{3.50}$$

And the RBLP solution is

$$\boldsymbol{a}^{(\boldsymbol{k+1})} = -(\boldsymbol{C}^{**})^{-1}\boldsymbol{c}^{**} \tag{3.51}$$

Hence, the algorithm simply reweights the residuals $e_n(\boldsymbol{a})$ by a proper weighting function $W(e_n(\boldsymbol{a}))$ and generate a weighted covariance matrix $\boldsymbol{C}^{**}$ and a weighted correlation vector $\boldsymbol{c}^{**}$ then solve for $\boldsymbol{a}$ by matrix inversion.

3.3.2 Solving the RBLP problem by Weighted Least Absolute Value Minimization [11]

The LPC's in this method are found so that they minimize a weighted absolute value of the error. Thus, $\boldsymbol{a}$ is the solution to the following $\ell 1$ minimization problem:

$$\hat{\boldsymbol{a}} = \min_{\boldsymbol{a}} \sum_{n=p+1}^{N} w(n)\,|e_n(\boldsymbol{a})| \tag{3.52}$$

where $w(n)$ is a Hamming window weight.

This problem is set as a linear program that is solved by the simplex method described in [13].

3.3.3 Stability of the RBLP Algorithms

As mentioned in Subsection 3.1.1.a, the autocorrelation method guarantees stability of the resultant system [6]. RBLP procedures, however, do not assure stability and hence require stability checks. If the RBLP algorithm produces an unstable LP filter with $A(z)$ having roots outside the unit circle, then the procedure can be stopped, and the stable preliminary LP filter is then used in the synthesis filter.

4.0 COMPRESSIVE SAMPLING

Compressive Sampling (CS), also known as Compressed Sensing is an emerging technique for data acquisition that promises sampling a sparse signal from a far fewer number of measurements than its dimension. It was motivated by the desire of sampling and compression simultaneously, instead of spending too much effort on sampling than throwing away most of what is sampled in the compression stage. The technique was introduced by David L. Donoho in 2006 [14] and has attracted attention ever since. In 2008 Emmanuel J. Candes and Michael B. Wakin [15] fully introduced the developed method to the signal processing society as a scheme that offers more efficient transmission, reception, and storage of data.

Compressed sensing is based on the idea that one can sufficiently capture all the information in a sparse signal by sampling only part of the signal using a sampling domain that is incoherent to the signal representation domain. A block diagram of the compressive sampling technique is shown in Figure 13 below; later sections of this chapter will fully explain each process of every block.

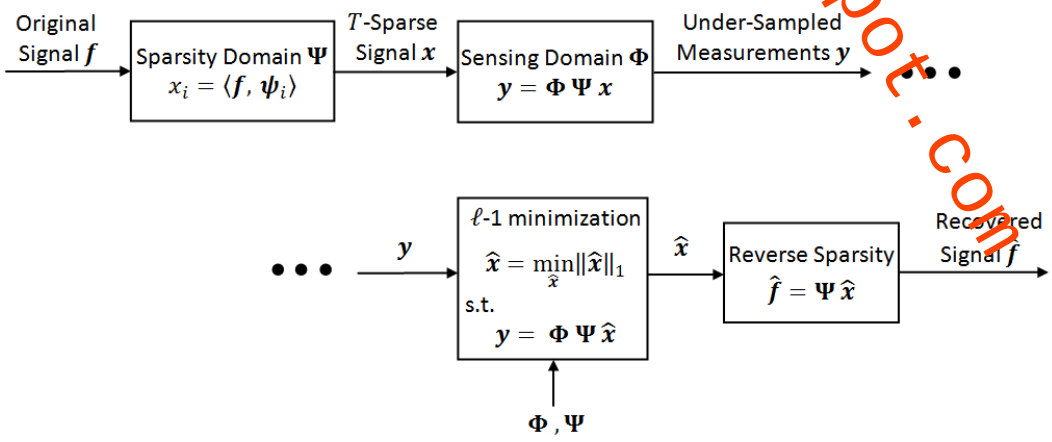

Figure 13. Block diagram of the compressive sampling procedure

This chapter is organized as follows. Section 4.1 introduces the concept and the mathematical representation of sparsity and incoherence as the two basic concepts of compressive sampling. The compressed sensing problem and the algorithms used to solve it are addressed in detail in Section 4.2; followed by a discussion about compressive sampling optimality in Section 4.3.

4.1 SPARSITY AND INCOHERENCE

Compressive sampling relies on two important properties; one is related to the signal that is about to be sampled (sparsity) and the other is related to the sampling domain (incoherence). The compressed sensing method is interested in highly sparse signals and highly incoherent sampling domains [16]. We now set the definition and mathematical representation of sparsity and incoherence.

4.1.1 Sparsity

Signals that are mostly populated with zeros and have a small number of non-zero components are called sparse signals. An example of a sparse signal is the multi-pulse excitation signal discussed in Section 3.2; where the excitation signal is mostly zero with few non-zero pulses. It was discussed in the previous chapter that such an excitation signal is sent over the transmission channel by quantizing and sending only the amplitudes and locations of the non-zero pulses. Sparsity hence allows efficient compression, interpretation, estimation and computation and thus plays a key role in compressive sampling.

Mathematically speaking, let $\boldsymbol{f} \in \mathbb{R}^N$ be an N-dimensional signal that is represented in a proper orthonormal basis $\boldsymbol{\Psi} = \{\boldsymbol{\psi}_1, \boldsymbol{\psi}_2, \dots, \boldsymbol{\psi}_N\}$, (i.e. $\boldsymbol{\psi}_i$'s are orthogonal unit vectors)

$$f(t) = \sum_{i=1}^{N} x_i \psi_i(t) \quad , \quad t = 1,2,\dots,N \tag{4.1}$$

where $\boldsymbol{x}$ is the coefficients sequence of $\boldsymbol{f}$ and $\boldsymbol{\psi}_i$ is an $N \times 1$ column.

Equivalently,

$$\boldsymbol{f} = \boldsymbol{\Psi}\boldsymbol{x} \quad \text{and} \quad x_i = \langle \boldsymbol{f}, \boldsymbol{\psi}_i \rangle \tag{4.2}$$

If we define $\boldsymbol{f}_T = \boldsymbol{\Psi}\,\boldsymbol{x}_T$,

$$f_T(t) = \sum_{i=1}^{N} x_{T_i} \psi_i(t) \quad t = 1,2,\dots,N \tag{4.3}$$

where, $\boldsymbol{x}_T$ is the vector of coefficients (x_i) with all but the T largest coefficients set to zero. In other words, $\boldsymbol{x}_T$ is a sparse vector with only T non-zero elements; $\boldsymbol{x}_T$ is called T-sparse.

If $\boldsymbol{x}$ is well approximated by $\boldsymbol{x}_T$, then the error $\|\boldsymbol{x} - \boldsymbol{x}_T\|_{\ell_2}$ is small. However, $\boldsymbol{\Psi}$ is an orthonormal basis and hence,

$$\|\boldsymbol{f} - \boldsymbol{f}_T\|_{\ell_2} = \|\boldsymbol{x} - \boldsymbol{x}_T\|_{\ell_2} \tag{4.4}$$

Therefore $\boldsymbol{f}$ is well approximated by $\boldsymbol{f}_T$ $(\boldsymbol{f} \approx \boldsymbol{f}_T)$. This means that if $\boldsymbol{x}$ is sparse, one can throw away a large fraction of the coefficients (x_i) without much loss in $\boldsymbol{f}$.

An example where the loss in $\boldsymbol{f}$ is relatively small is shown in Figure 14, which shows a very dense audio clip in the time domain (a) and its sparse representation in the Discrete Cosine Transform (DCT) basis (b). Since the largest DCT coefficients carry most of the energy [17], only the coefficients corresponding to 97% of the signal's energy are kept and the rest are discarded; which is achieved by zeroing out the smallest 83% of the DCT coefficients. Figure 14 (c) shows the audio clip reconstructed from the largest 17% of the DCT's.

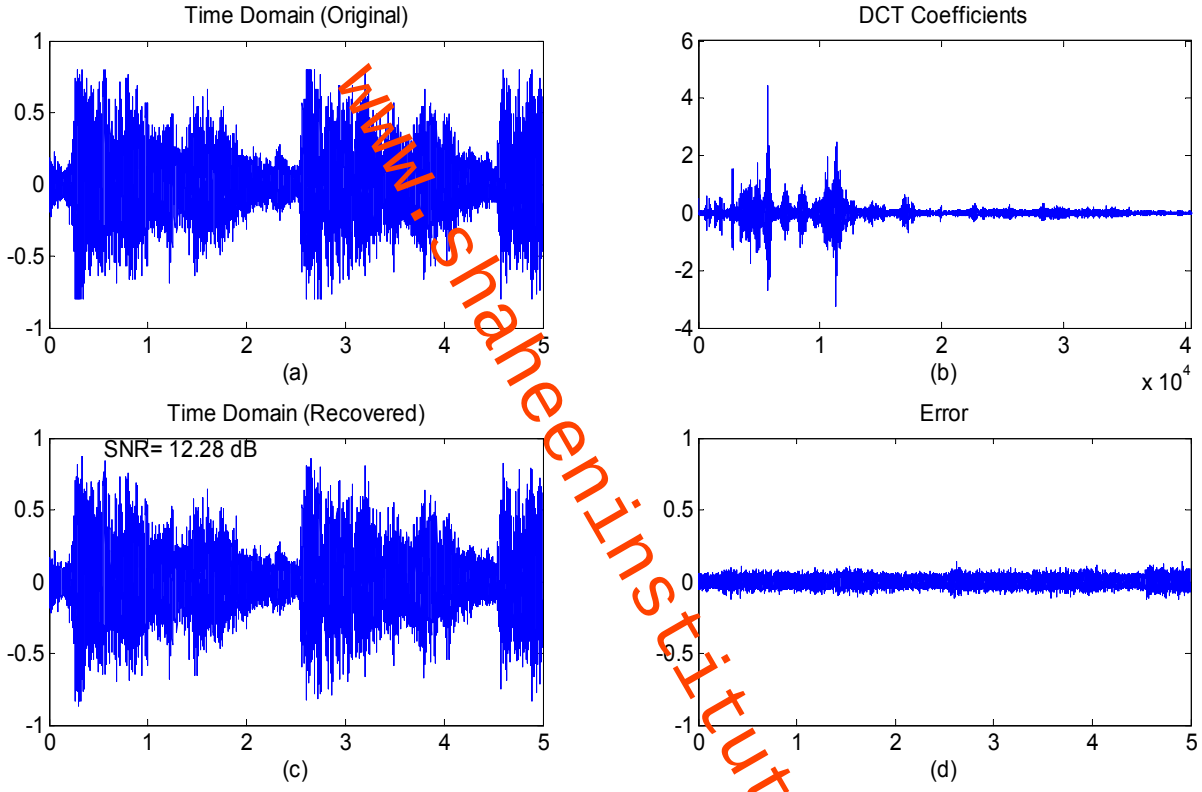

Figure 14. **Sparse signal recovery. (a) Original signal, 5 sec Audio clip of Handel's Messiah (b) The discrete cosine transform coefficients of the signal (c) The reconstructed audio clip from the 17% largest DCT's (d) The error signal**

Hence, a simple method for data compression would be to compute $\boldsymbol{x}$ from $\boldsymbol{f}$ then (adaptively) encode the locations and amplitudes of the T most significant coefficients. This principle actually underlies many modern lossy coders [15]; however, compressive sampling is a different concept where the sparsity of the signal has significant bearings on the acquisition process itself; sparsity determines how efficiently one can (non-adaptively) acquire a signal [15].

Not all signals are sparse by their nature; however most signals are sparse when expressed in the proper basis. Therefore it is very important to find the (right) basis where most signals of the same nature are sparse in order to be able to perform compressed sensing independently on the signal.

4.1.2 Incoherent measurement basis

Incoherence extends the duality between time and frequency expressed in the uncertainty principle to the duality between the signal's sparse representation and the domain where it is sampled [15]. Just as a Dirac or a spike in the time domain is spread out in the frequency domain, a signal that has a sparse representation in $\boldsymbol{\Psi}$ must be spread out in the domain $\boldsymbol{\Phi}$ in which it is acquired. Put differently, incoherence says that unlike the signal of interest, the sampling waveform has an extremely dense representation. One good example of a sparse/dense pair is sampling a sequence of Dirac pulses (very sparse) in a sinusoidal basis (very dense).

In order to take m measurements of a vector $\boldsymbol{f} \in \mathbb{R}^N$ we sample $\boldsymbol{f}$ in the sampling domain $\boldsymbol{\Phi}$, where $\boldsymbol{\Phi} = \{\boldsymbol{\varphi}_1, \boldsymbol{\varphi}_2, \dots, \boldsymbol{\varphi}_N\}$ and $\boldsymbol{\varphi}_i$ is an $m \times 1$ column. The measurements signal is therefore defined as:

$$y(k) = \sum_{i=1}^{N} f(i)\varphi_i(k), \qquad k = 1, 2, \dots, m \tag{4.5}$$

The coherence between the representation matrix $\boldsymbol{\Psi}$ and the measurements matrix $\boldsymbol{\Phi}$ is defined as [18]

$$\mu(\boldsymbol{\Phi}, \boldsymbol{\Psi}) = \sqrt{N} \max_{1 \le k,j \le N} \left|\langle \boldsymbol{\varphi}_k, \boldsymbol{\psi}_j \rangle\right| \tag{4.6}$$

If $\boldsymbol{\Phi}, \boldsymbol{\Psi}$ are normalized such that $\|\boldsymbol{\Phi}_{m\times N}\|_2 = \|\boldsymbol{\Psi}_{N\times N}\|_2 = 1$

Then $\langle \boldsymbol{\varphi}_k, \boldsymbol{\psi}_j \rangle \le 1 \quad \Rightarrow \quad \mu \le \sqrt{N}$

However, $\sum_{j=1}^{N} \left|\langle \boldsymbol{\varphi}_k, \boldsymbol{\psi}_j \rangle\right|^2 = 1 \quad , \quad k = 1, 2, \dots, N$

thus $\langle \boldsymbol{\varphi}_k, \boldsymbol{\psi}_j \rangle \ge \frac{1}{\sqrt{N}} \quad \Rightarrow \quad \mu \ge 1$

$$\therefore \quad \mu(\boldsymbol{\Phi}, \boldsymbol{\Psi}) \in [1, \sqrt{N}] \tag{4.7}$$

As discussed in the next section, the smaller the coherence between $\boldsymbol{\Phi}, \boldsymbol{\Psi}$ the fewer measurements are taken by CS and hence the term "incoherence" is used.

Random matrices are widely used as sampling bases $\boldsymbol{\Phi}$ in CS applications; that is because CS is concerned with high incoherence and random matrices are largely incoherent with any fixed basis $\boldsymbol{\Psi}$ [15]. White Gaussian or uniform noise thus make good sampling bases for CS [19] and are widely used as the CS measurement basis.

Now that the foundation of CS is laid, we move on to formulating and defining the CS problem.

4.2 THE COMPRESSIVE SAMPLING PROBLEM

The compressive sampling problem asks two basic questions, how many measurements are needed to fully capture the information in the signal? and what methods are used to recover the data from the undersampled measurements?

The first question was answered by Candes, Romberg, and Tao [20] who suggested that to capture the information in the signal with a probability of $1 - N^{-M}$, where M is a positive constant, one needs to take a number of measurements m that is proportional to both the sparsity level T and a log factor of the signal dimension N.

$$m \geq \text{Const} \;\; M \;\; T \;\; \log(N) \tag{4.8}$$

This result was then enhanced to [15]

$$m \geq C \;\; \mu^2(\boldsymbol{\Phi}, \boldsymbol{\Psi}) \;\; T \;\; \log(N) \tag{4.9}$$

where C is some positive constant and $\mu(\boldsymbol{\Phi}, \boldsymbol{\Psi})$ is the coherence.

Further simplifications are made when the sensing basis is highly incoherent with the representation basis, e.g. when taking $\boldsymbol{\Phi}$ as white noise, then the coherence term can be absorbed in the constant C, and m can be simplified to

$$m \geq C \ T \ \log(N) \tag{4.10}$$

The second question was tackled by many ways and the literature is rich with algorithms that are developed to recover highly incomplete information. The methods of finding the solution to the CS problem generally fall into two classes, methods which use linear programs to recover the data (basis pursuit) and methods that use second order greedy algorithms (orthogonal matching pursuit).

4.2.1 Solving the CS problem using basis pursuit algorithms (ℓ-1 minimization)

The ℓ-1 minimization approach, also known as Basis Pursuit (BP) algorithm, is one major approach to solve the CS problem and was presented in the early CS work as the best algorithm for sparse signal recovery.

THEOREM 1 [15], [18]

Let $\boldsymbol{f} \in \mathbb{R}^N$ be an N dimensional signal that is T-sparse in some basis $\boldsymbol{\Psi}$ (i. e. $\boldsymbol{f} = \boldsymbol{\Psi}\,\boldsymbol{x}$ and $\boldsymbol{x}$ is T-sparse). Collect m measurements independently and randomly in a white Gaussian domain $\boldsymbol{\Phi}$ such that

$$m \geq C \ T \ \log(N/T) \tag{4.11}$$

where C is some positive constant.

Then it is possible to reconstruct every T-sparse signal $\boldsymbol{x}$ (and hence recover $\boldsymbol{f}$) with a probability exceeding $1 - e^{-cm}$ (where c is a constant different from C) by solving the following convex optimization problem,

$$\hat{\boldsymbol{x}} = \min_{\hat{\boldsymbol{x}}} \|\hat{\boldsymbol{x}}\|_1 \quad \text{subject to} \quad \boldsymbol{y} = \boldsymbol{\Phi}\,\boldsymbol{\Psi}\,\hat{\boldsymbol{x}} \quad \text{and} \quad \hat{\boldsymbol{f}} = \boldsymbol{\Psi}\,\hat{\boldsymbol{x}} \tag{4.12}$$

where $\boldsymbol{y}$ is the vector of sampled measurements $\boldsymbol{y} = \boldsymbol{\Phi}\,\boldsymbol{f}$ and $\hat{\boldsymbol{f}}$ is the reconstructed signal.

An important detail in the BP approach is that a particular choice of the Gaussian measurement matrix $\boldsymbol{\Phi}$ succeeds to recover every T-sparse signal with high probability [21].

Below are two examples of the ℓ-1 minimization recovery results. Figure 15 illustrates example I, where a sparse signal of length 512 with only 20 non-zero $+/-$ perfect unit pulses placed randomly is recovered using compressive sampling. The signal was successfully recovered from only 120 measurements taken randomly in a white Gaussian basis. The obtained SNR is overwhelmingly high, 218.41 dB, for a compression factor of 0.23.

Figure 16 illustrates example II where the signal is a 512 vector with T=128 non-zero elements that are not perfect unit pulses, rather they are drawn randomly from a normal distribution and placed at random locations. The results show an exact recovery with SNR of 179.9 dB when taking only 384 measurements. This is a case where the number of measurements is only 3 times the sparsity level T (less than $T \log N$), yet the recovery is successful with an overwhelmingly high SNR.

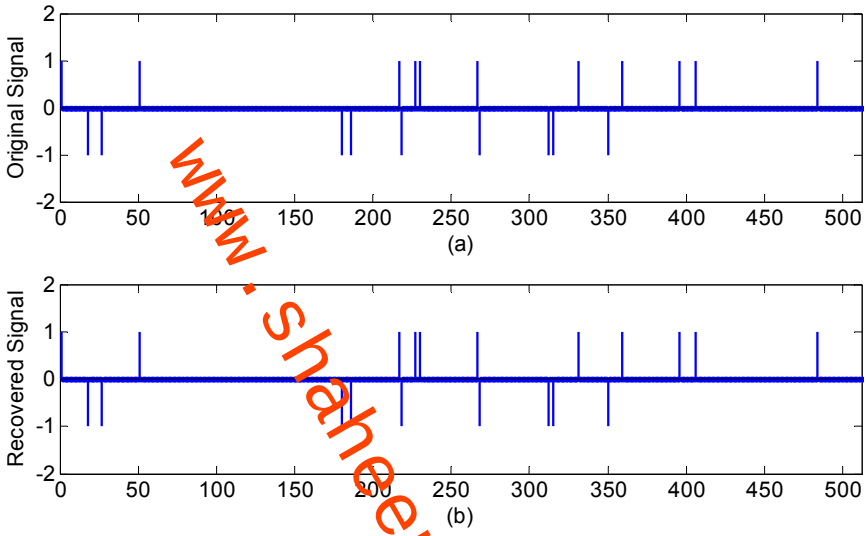

Figure 15. **Sparse signal recovery using $\ell 1$-minimization - example I**
(a) The original sparse signal (b) The signal exactly recovered by CS collecting only 120 measurements

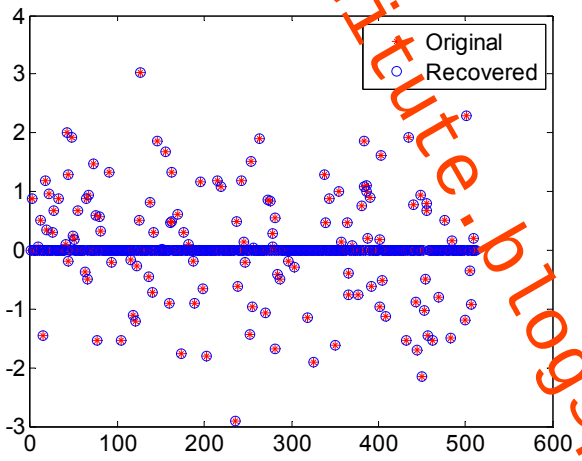

Figure 16. **Sparse signal recovery using $\ell 1$-minimization - example II**

The Basis Pursuit algorithm guarantees recovery of the sampled signal with a very high probability of success. However, since the algorithm solves an iterative linear program, the computation expense of solving the CS problem using BP algorithms can be high

4.2.2 Solving the CS problem using orthogonal matching pursuit algorithms

As mentioned above, the ℓ-1 minimization approach can be expensive in terms of time and computation especially for large data; Orthogonal Matching Pursuit (OMP) algorithm was therefore introduced (along with some other greedy algorithms) to minimize the time and computation cost of solving the CS problem.

THEOREM 2 [21]

Given a signal $\boldsymbol{f} \in \mathbb{R}^N$ that is T-sparse in $\boldsymbol{\Psi}$, fix $\delta \in (0\ ,\ 0.36)$ and choose m such that,

$$m \geq C\ T\ \log(N/\delta) \tag{4.13}$$

Then $\boldsymbol{f}$ can be recovered from m measurements drawn independently from a standard Gaussian distribution with a probability that exceeds $1 - 2\delta$ by solving the following ℓ-2 minimization problem:

$$\hat{\boldsymbol{x}} = \min_{\hat{\boldsymbol{x}}} \|\boldsymbol{y} - \boldsymbol{\Phi}\,\boldsymbol{\Psi}\,\hat{\boldsymbol{x}}\|_2 \ \text{ subject to } \ \|\hat{\boldsymbol{x}}\|_0 = T$$
$$\text{and} \quad \hat{\mathbf{f}} = \boldsymbol{\Psi}\,\hat{\boldsymbol{x}} \tag{4.14}$$

The idea behind the OMP algorithm is to pick columns in a greedy fashion [21]. Let the sensing matrix $\boldsymbol{\Phi} = \{\boldsymbol{\varphi}_1,\ \boldsymbol{\varphi}_2, \ldots,\ \boldsymbol{\varphi}_N\}$ where $\boldsymbol{\varphi}_i$ is an $m \times 1$ column. At each of the T iterations, select the column $\boldsymbol{\varphi}_i$ that is most strongly correlated with $\boldsymbol{y}$; then subtract off its contribution to $\boldsymbol{y}$ and iterate on the subtraction residual. It is actually a search process very similar to the one discussed in Section 3.2 where the number of pulses searched for is fixed to T. The greedy algorithm as introduced by Tropp and Gilbert in [21] can be summarized as:

1. Initialize the residuals $\boldsymbol{r}_0 = \boldsymbol{y}$, the index set $\boldsymbol{\Lambda}_0 = \emptyset$, the matrix of chosen atoms $\boldsymbol{\Phi}_0 = \emptyset$, and the iteration counter $t = 1$.

2. Find the index $\lambda_t = \arg\max_{j=1,\ldots,N} \left|\langle \boldsymbol{r}_{t-1}, \boldsymbol{\varphi}_j \rangle\right|$.

3. Augment the index set and the matrix of chosen atoms:

$$\boldsymbol{\Lambda}_t = \boldsymbol{\Lambda}_{t-1} \cup \{\lambda_t\} \text{ and } \boldsymbol{\Phi}_t = \left[\boldsymbol{\Phi}_{t-1} \quad \boldsymbol{\varphi}_{\lambda_t}\right].$$

4. Solve the basic least squares problem to find a new estimate: $\boldsymbol{s}_t = \arg\min_{\boldsymbol{s}} \|\boldsymbol{y} - \boldsymbol{\Phi}_t \boldsymbol{s}\|_2$

5. Calculate the new data approximation and the new residuals:

$$\boldsymbol{a}_t = \boldsymbol{\Phi}_t \boldsymbol{s}_t \text{ and } \boldsymbol{r}_t = \boldsymbol{y} - \boldsymbol{a}_t$$

6. Increment t and go to step 2 if $t < T$.

7. The estimate $\hat{\boldsymbol{x}}$ has non-zero elements only at the indices listed in $\boldsymbol{\Lambda}_T$. The value of $\hat{\boldsymbol{x}}$ in component λ_j equals the jth component of $\boldsymbol{s}_t$.

Figure 17 illustrates example I, the same example in Figure 15, where a 512 samples signal is recovered from only 120 measurements but this time using OMP algorithm.

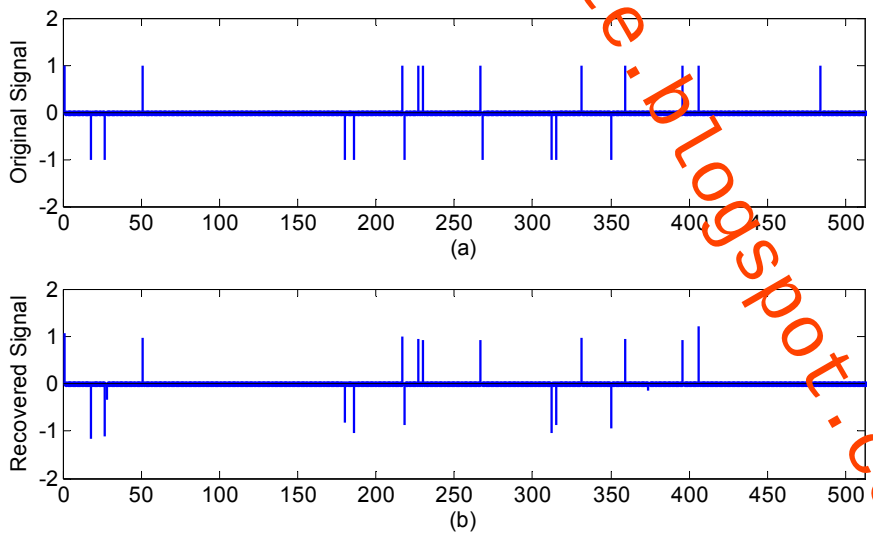

Figure 17. **Sparse signal recovery using OMP algorithm, example I. (a) The original sparse signal (b) The signal recovered by CS collecting only 120 measurements**

Observing Figure 17.b, one can notice that the algorithm fails to recover spikes 12 and 20; further, the amplitudes of other recovered spikes are notably different than the originals. This degrades the SNR to 9.32 dB compared to the 218 dB obtained using BP algorithm.

The run time and computation cost of the OMP algorithm are far less than for the BP approach; however, OMP is weaker than BP for many reasons [21]:

- More samples are needed for OMP than for BP for a successful recovery.
- The probability of success is smaller for OMP than the one for the BP.
- The quantifiers are ordered differently, whereas OMP recovers each sparse signal with high probability but with high probability fails to recover all sparse signals, BP shows that a single set of random measurement vectors can be used to recover all sparse signals. In other words, OMP solution is non-uniform in contrast to BP solution.

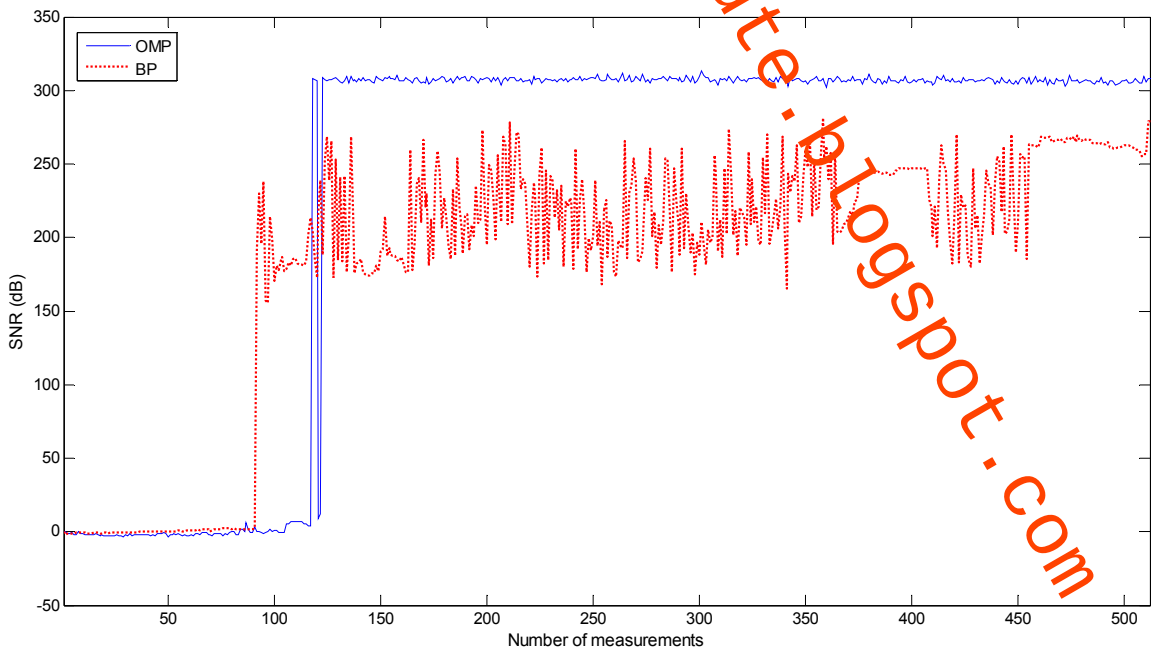

Figure 18. BP vs. OMP performance for the signal of example I

Figure 18 illustrates the above points on example I. The plots compare the performance of OMP to that of BP using the same signal and the same sensing matrix. It is clear that both approaches result in an overwhelmingly high SNR (higher than 150 dB) when taking enough measurements. It is also clear that OMP needs more samples, to achieve its high SNR, than the number of measurements BP needs; which makes the performance of BP better than the performance of OMP especially at lower compression factors. However, in some applications high performance is compromised by the run time and the computation cost making OMP algorithms some times more practical.

4.3 OPTIMALITY OF COMPRESSIVE SAMPLING TECHNIQUES

CS as mentioned above, assures the recovery of undersampled signals with an overwhelming high probability if the signal of interest is sparse in some basis $\boldsymbol{\Psi}$, sampled in a random domain $\boldsymbol{\Phi}$, if the number of measurements m is chosen according to equation (4.10) and if the ℓ-1 minimization approach is used. However there exist some very extreme cases where all the conditions are satisfied and yet CS fails to recover the signal with high probability.

One very simple example is a signal $\boldsymbol{x}$ that has only one non-zero element. In this case, with high probability, the ℓ-1 minimization approach returns a signal $\hat{\boldsymbol{x}}$ that produces the same measurements signal, $\boldsymbol{y} = \boldsymbol{\Phi}\,\hat{\boldsymbol{x}} = \boldsymbol{\Phi}\,\boldsymbol{x}$, but has a lower ℓ-1 norm than $\boldsymbol{x}$. Figure 19 shows an experiment that illustrates the CS failure of recovering a single pulse signal. The signal in Figure 19 has a single unit pulse and 511 zero elements. The number of measurements was calculated according to (4.10), $m \geq C\ T\ \log(N)$, with $C = 2$, $T = 1$ and $N = 512$. 13 measurements were taken; the experiment was run 100 times and 7 attempts to recover the signal failed.

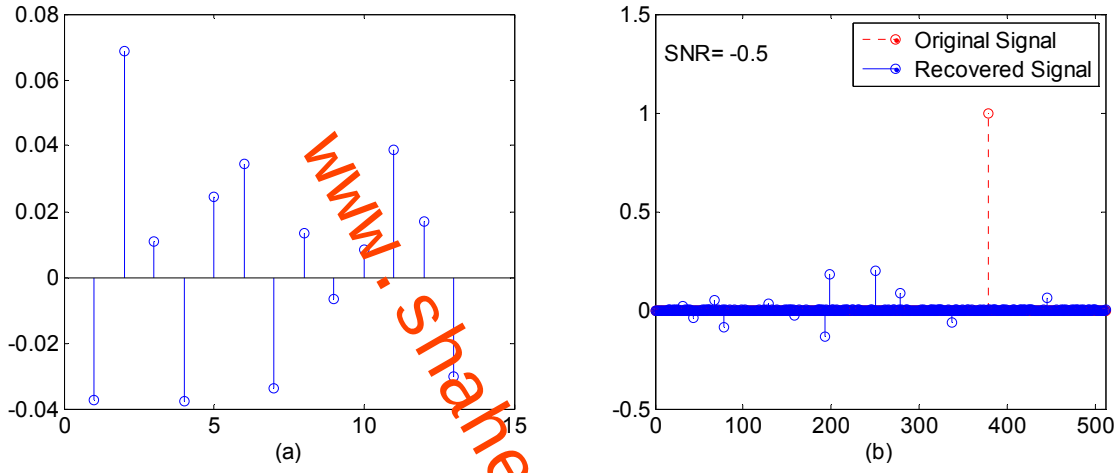

Figure 19. **CS failure to recover a single spike signal (a) Sampled signal (b) Unsuccessful recovery The ℓ-1 norm of the recovered signal is 0.9898 (less than 1)**

The single pulse signal however is not the only high sparse signal that cannot be recovered, with high probability, using CS. Figure 20 shows an experiment where the sparsity level varies from 1 to 20 for signals with different lengths. For each signal, $m = T\ \log(N)$ measurements are taken and the probability of success was calculated out of 1000 trials.

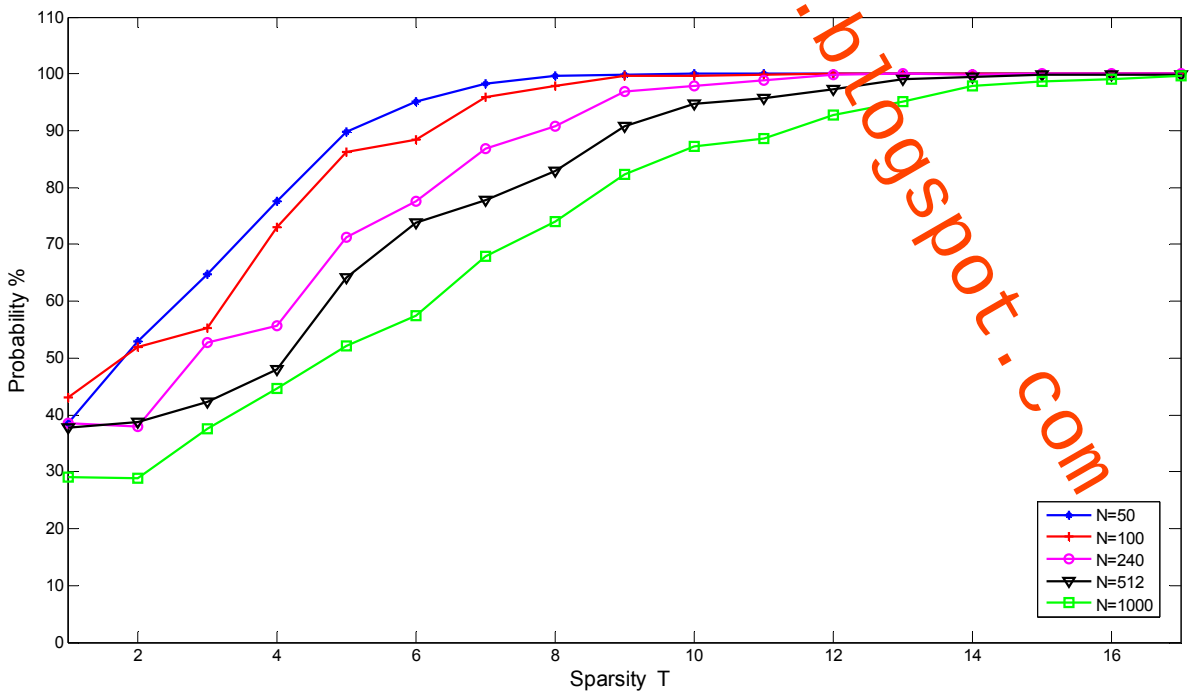

Figure 20. **Probability of successfully recovering signals with different lengths N and sparsity T when taking $m = \lceil T\log(N) \rceil$ measurements**

The plots in Figure 20 indicate that there is a lower bound for the sparsity in order to successfully recover the signal with high probability if the same constant C in equation (4.10) is used; the plots further show that this lower bound is proportional to the signal length.

Although the probability of success is so low for some low sparsity signal, these signals can still be recovered only if the constant C is increased so that more measurements are taken. Which means that C varies depending on the signal's length and sparsity; i.e. higher C is needed for lower sparsity than for higher sparsity. In other words, there is a lower bound on the number of measurements for CS to successfully recover the signal (with high probability).

5.0 COMPRESSIVE SAMPLING OF SPEECH SIGNALS

As described in the introduction (Chapter 1), the purpose of our work is to take advantage of the promising technique of compressive sampling to develop an algorithm that applies compressive sampling to speech signals in order to transmit them in lower bit rates than the existing algorithms or with the same bit rates but with better speech quality.

As stated in Chapter 4, CS techniques require the signal to be sparse in some basis. Speech signals are very dense in the time domain; further, the domains in which they are sparse are neither clear nor straight forward due to the different speech production and classification schemes. If we consider the production mechanism described in Chapter 2 and the classification of the speech into only voiced and unvoiced signals; we can consider the voiced signal residuals as a suitable sparse domain. As shown in Chapter 3, the speech residuals are sparse for voiced sounds and are dense but have a random nature for unvoiced sounds. We can thus take advantage of these properties and try to apply CS to the residual signal, obtained from LP analysis, and then reconstruct the speech signal by passing the CS recovered residuals through the LP synthesis filter.

In this chapter, the compressive sampling technique is applied to the speech signal. CS will be implemented on both clean and noisy speech samples for multiple male and female speakers. Since, as discussed in Section 3.3, robust linear prediction techniques result in better approximations for the LPC's; we will apply CS on speech residuals obtained from both conventional and robust linear prediction approaches.

This chapter is divided into five sections; Section 5.1 will explain in detail the implementation procedure. The following sections will introduce some implementation results.

In Section 5.2, CS is applied on speech residuals obtained by analyzing the speech signal using LPC's found by the Conventional Linear Prediction (CLP) analysis; the speech signal is then synthesized using the LPC's and the speech signal is studied and compared to the original one. In Section 5.3, the same steps as in Section 5.2 are preformed; the only difference is that the LPC's are obtained by analyzing the speech signal using Robust Linear Prediction (RBLP) analysis. Section 5.4 provides a comparison between the two approaches used in Sections 5.2 and 5.3. While Section 5.5 addresses the issue of finding the best thresholding method.

5.1 COMPRESSIVE SAMPLING IMPLEMENTATION PROCEDURE

The process explained in this section is used throughout the implementation sections unless different steps are stated; all implementations are performed using MATLAB.

As stated above, CS will be performed on the residuals signal, not on the speech directly. The residual signal of a voiced speech frame as plotted in Figure 12.a is a quasi-periodic signal with major large pulses at the pitch period time intervals and some other smaller pulses in between. Makhoul and Berouti [22] showed that 41% of the residuals amplitudes are almost zero. This means that the residuals signal is nearly sparse. They further showed that almost 88% of the residual signal is of very small amplitudes. Thus, one can threshold these small amplitudes and set them to zero to increase the sparsity of the residual signal without terribly affecting the synthesized speech signal.

CS will be performed on the thresholded residuals, for different threshold levels, and the CS recovered residuals will be compared to the original residuals (before thresholding); finally a

speech signal is synthesized using the LPC's and the recovered residuals and will be compared to the original speech signal.

The first stage of the implementation is to segment the speech signal into 30 ms frames then find the 12th order linear prediction coefficients for each Hamming windowed speech frame. The second stage is analyzing the speech frames to find the speech residuals and the residuals variance (power).

As mentioned above, the residual signal is thresholded for different levels where the threshold level is responsible for determining the compression factor; we call the thresholded residuals signal $\boldsymbol{x}$ to stay in the same notation used in Chapter 4. The sparsity level T is set to be the total number of non-zero elements left in the residual signal after thresholding. The number of measurements per frame m is found using Equation (4.10) with the constant C set to 1.

$$m = \lfloor T \log(N) \rfloor \tag{6.1}$$

where $\lfloor \cdot \rfloor$ denotes the floor function.

Note: This approach provides dynamic computation; frames with a large number of pulses will need a large number of sensed values and thus a higher computational effort compared to frames with few pulses.

The compression factor CF is calculated as the sum of the measured samples m for all the frames divided by the size of the original speech signal; i.e. lower compression factors indicate more compression.

$$CF = \frac{m_1 + m_2 + \cdots + m_{n_f}}{n_f N} \tag{6.2}$$

where m_i denotes the number of measurements taken from the ith frame; and n_f is the total number of frames.

Now that the number of measurements is known, we can apply compressive sampling. A matrix $\boldsymbol{\Phi}^*$ of size $N \times N$ is formed of orthogonal random vectors drawn independently at random from a zero-mean unit-variance normal distribution and. For each frame, the same matrix $\boldsymbol{\Phi}^*$ is truncated to form the sensing $m \times N$ matrix $\boldsymbol{\Phi}$ that is used to take m measurements from the thresholded residuals signal; where the measurements signal $\boldsymbol{y}$ is defined as $\boldsymbol{y} = \boldsymbol{\Phi}\,\boldsymbol{x}$.

The measurements signal is then used to recover $\boldsymbol{x}$. Recovery is obtained by the ℓ-1 minimization procedure explained in Subsection 4.2.1. The minimization is carried out as a linear program and MATLAB's `linprog` is used to find the solution. The recovered residuals signal $\hat{\boldsymbol{x}}$ is then used to synthesize the speech signal using the LPC's found at the beginning.

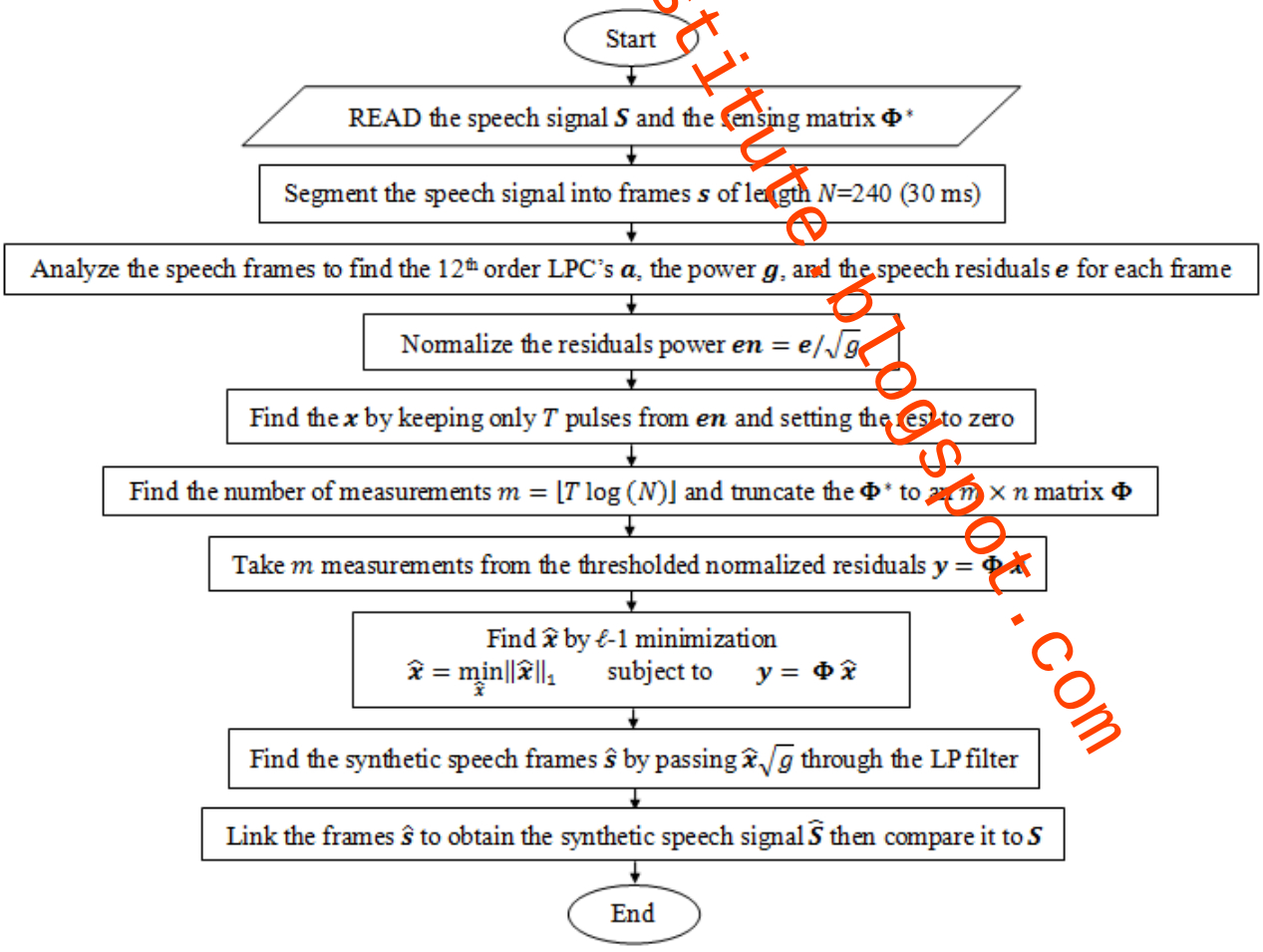

Figure 21. Compressive sampling implementation flowchart

5.2 COMPRESSIVE SAMPLING ON CLP RESIDUALS

In this section, CS will be applied to the residuals obtained by analyzing the speech signal using LPC's found by the classical linear prediction analysis. The LPC's are found using MATLAB's `lpc` function which uses the autocorrelation method and the Levinson-Durbin recursion explained in detail in Subsection 3.1.1.

Figure 22 below shows an SNR plot with different compression factors for four different speakers speaking different phrases. A list with the spoken phrases can be found in the appendix. The plots show the improvement of the SNR with increasing the compression factor. At a compression factor of 1 the SNR grows to values over 250 dB indicating exact recovery. The plots also indicate that the SNR obtained at the same compression factor is lower for female speakers than for male speakers, even if the environment for the female speaker is noise free. This is expected since the pitch of a female speech is higher than for a male's; and thus the residual signal of female speech has more non-zero elements which means higher T and thus higher m is needed. Therefore for the same compression factor, female speakers require more thresholding than male speakers which results in a lower SNR. For male speakers on the other hand, the SNR does not improve much (only 1 to 2 dB higher) when speaking in a noise free setting; however, listening tests show better quality synthesized speech when recording in a noise free environment than a noisy one even if the SNR is higher for the noisy environment speech. This makes sense because the coding is parametric, not waveform coding; and hence SNR is not the best criteria to judge the speech quality in this case listening tests are better criterion.

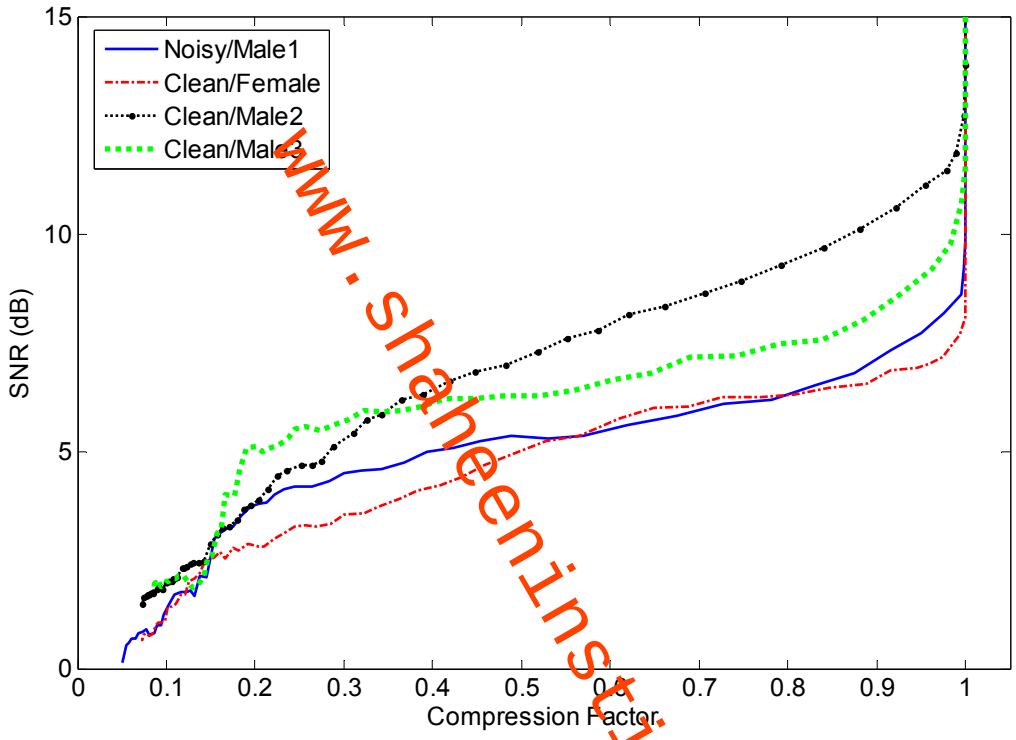

Figure 22. **CS recovery performance (SNR) for residuals obtained using CLP**

The plots in Figure 22 represent the total SNR; that is the SNR of the full length original speech and the full length CS recovered speech signal. However, as mentioned earlier, the CS process is performed on the residuals signal on a frame by frame basis. Therefore the total speech SNR is not be a good indication of the success or failure of the CS process. In order to discuss the probability of failure that is present in the CS method, a plot with the CS process input and output is shown in Figure 23 for each frame. The original residuals are thresholded so that the compression factor is almost 0.3. The thresholded residuals are the input to the CS system where the recovered residuals are the output of the system. The stairs plot is the SNR (original residuals vs. recovered residuals) for each frame.

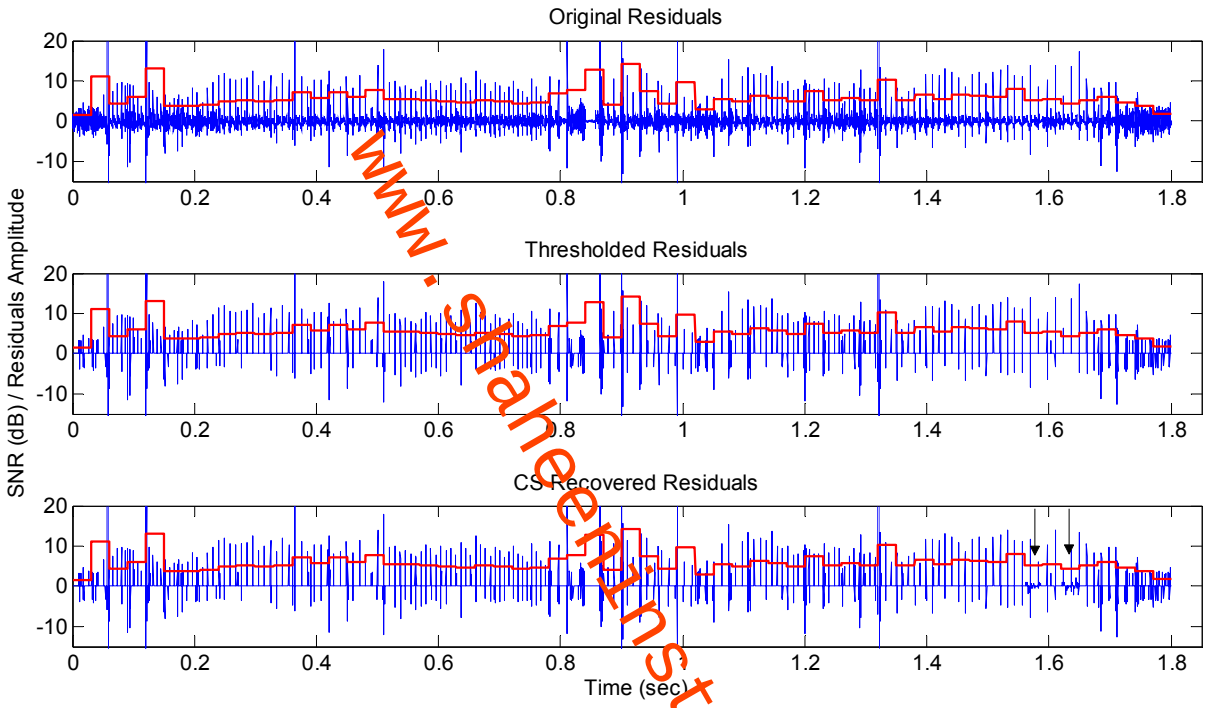

Figure 23. **Frame by frame comparison between original, thresholded, and recovered residuals (CLP) with a stair SNR (Original vs. Recovered) plot for Male3 speaker.**

The bottom graph of Figure 23 shows two frames (marked with the vertical arrows) where the CS system fails to recover the input signal despite meeting all the method's requirements. The failure simply falls in the small probability margin of failure. Investigating these two frames we note that the first frame has only 5 pulses in the thresholded frame and the second one has 6 pulses. Referencing Figure 20 in Chapter 4, the estimated probability of failure for signals with 5 and 6 pulses is almost 28% and 22% respectively; which is a high probability. Other than those two frames, the CS process successfully reconstructs its input with overwhelming SNR (as high as 200dB); the SNR stairs plot in Figure 23 is obtained when comparing the reconstructed residuals to the original ones.

Figure 23 also implies that the SNR for the recovered residuals is low in one of three cases. The first case is: the frame is unvoiced and this is expected since the unvoiced frames have low sparsity and thus CS is not the best choice to acquire them; however we don't care much

about noisy unvoiced frames since unvoiced sounds are already modifications of noise. The second case is: the residuals signal is poorly thresholded where the threshold level zeros out important pulses. Although we work with normalized power residuals so that the same threshold can be used for all the frames, some frames still have residuals with lower amplitudes (even at pitch periods) than other frames. These lower amplitude pulses fall below the threshold level, especially when the level is high (for lower compression factor), and hence important pulses are zeroed out. This issue can be resolved by decreasing the threshold level or using a different threshold for each frame. The third case is: bad luck, the CS process fails to recover the data simply because there is always a probability, even if small, of failure.

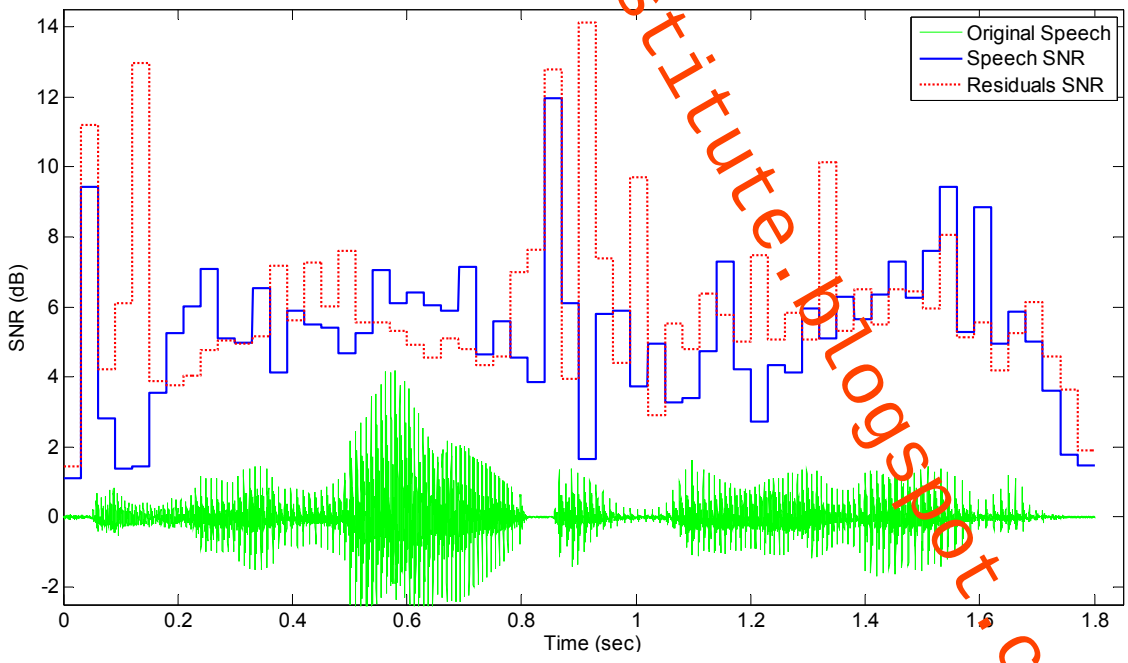

Figure 24. **Residuals and speech signal to noise ratio for each frame of the speech signal (The speech signal is amplified for the purpose of plotting)**

The SNR plots in Figure 23 are of the residuals signal, not the speech. Synthesizing the speech of the recovered residuals may increase the SNR, especially for voiced frames and if the synthesis filter coefficients best represent the speech production model; or/and it may decrease

the SNR especially for unvoiced frames. This can be seen in Figure 24, where the SNR of both the residuals and speech is plotted.

In the next Section, the performance of the CS process is investigated when the residuals are obtained by the robust linear prediction methods.

5.3 COMPRESSIVE SAMPLING ON RBLP RESIDUALS

In this section, the CS procedure is investigated for the case where the residuals are found by analyzing the speech using LPC's obtained from robust linear prediction methods. The BRLPC's are found by the iterative reweighted least squares algorithm explained in Subsection 3.3.1.

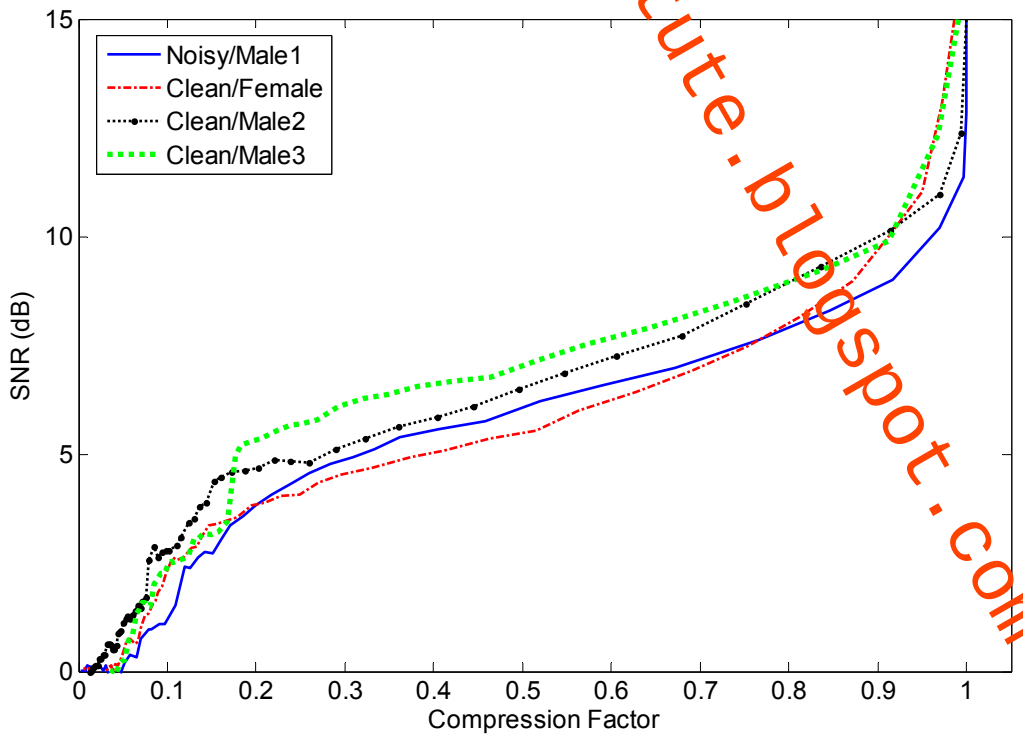

Figure 25. CS recovery performance (SNR) for residuals obtained using RBLP

Figure 25 above shows the SNR plots for the four speakers. As in Figure 22, the SNR improves with higher compression factors. The SNR for the female speaker is still lower than the SNR for male speakers for compression factors of 0.2 and higher. Also, listening tests show better speech quality for speech recorded in noise free settings.

The original, thresholded and recovered residuals at a compression factor of almost 0.3 are plotted in Figure 26 along with the residuals SNR. The bottom graph also shows three frames where the CS algorithm fails to reconstruct its input; the three frames are different from the two ones in the CLP case since the residuals are different. The failure also is a result of having a small number of pulses in the thresholded signals; 5, 3 and 5 pulses with a probability of failure of about 28%, 47% and 28% respectively. As in the CLP case, the plots show lower SNR for unvoiced frames compared to voiced ones.

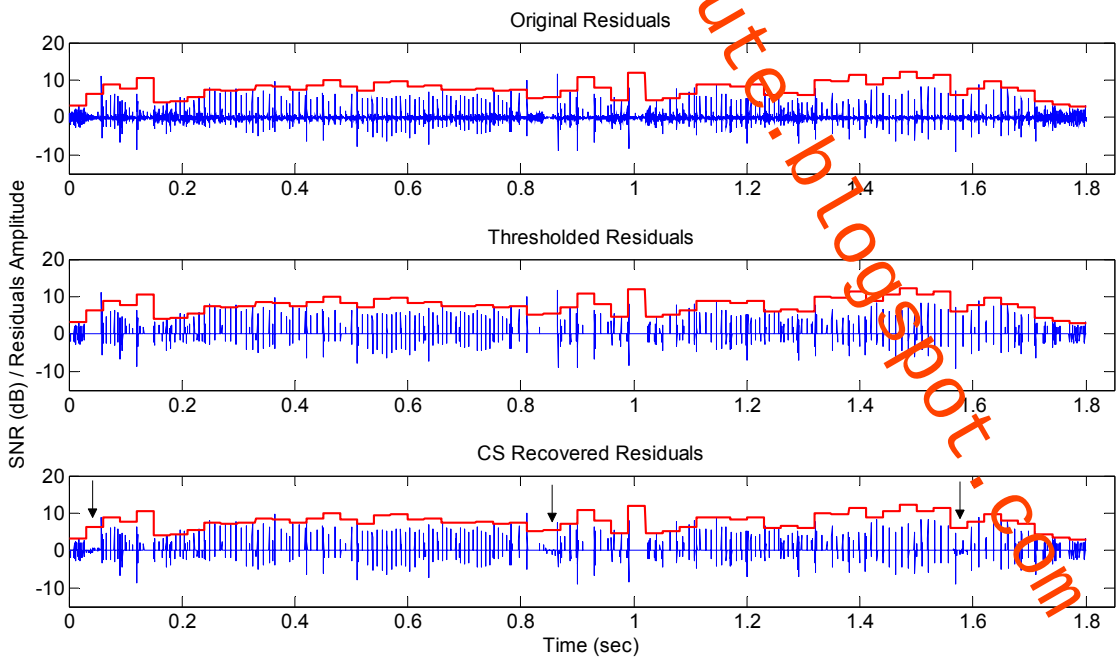

Figure 26. **Frame by frame comparison between original, thresholded, and recovered residuals (RBLP) with a stair SNR (Original vs. Recovered) plot for Male3 speaker**

In the next section, the output of the CS algorithm is compared for the case where the residuals are obtained from the CLP and the RBLP.

5.4 COMPRESSED SENSING ON CLP RESIDUALS VS. ON RBLP RESIDUALS

Now that the performance of the CS algorithm on residuals obtained by both the CLP and RBLP methods is investigated, it's time to compare the performance of both approaches.

Figure 27 shows SNR plots for speech sensed from residuals of CLP and RBLP approaches for four different speakers. SNR and listening tests indicate that for speech recorded in a noisy environment and for female speakers, the synthesized speech quality is better for the RBLP case than for the CLP. On the other hand, for speech recorded by male speakers in noise free environments, the speech quality is still better for the RBLP case; however, the SNR does not necessarily improve.

Again, Figure 27 shows a total speech SNR plots. Figures 28 and 29 below show a frame by frame SNR for both the reconstructed residuals and the corresponding synthesized speech for a male speaker in a noisy environment and another male speaker in a noise free environment, respectively, at a compression factor of 0.4. The plots show that for noisy speech, the SNR for speech synthesized from RBLP residuals is higher than that for CLP residuals for almost all the frames (voiced and unvoiced). While for clean speech it depends on the classification of the speech frame. Voiced frames demonstrate higher SNR for the RBLP case and unvoiced frames show higher SNR for the CLP case; while transitional frames show lower SNR for the RBLP case and frames of silence show severely low SNR that can be seen in both the RBLP and the CLP cases.

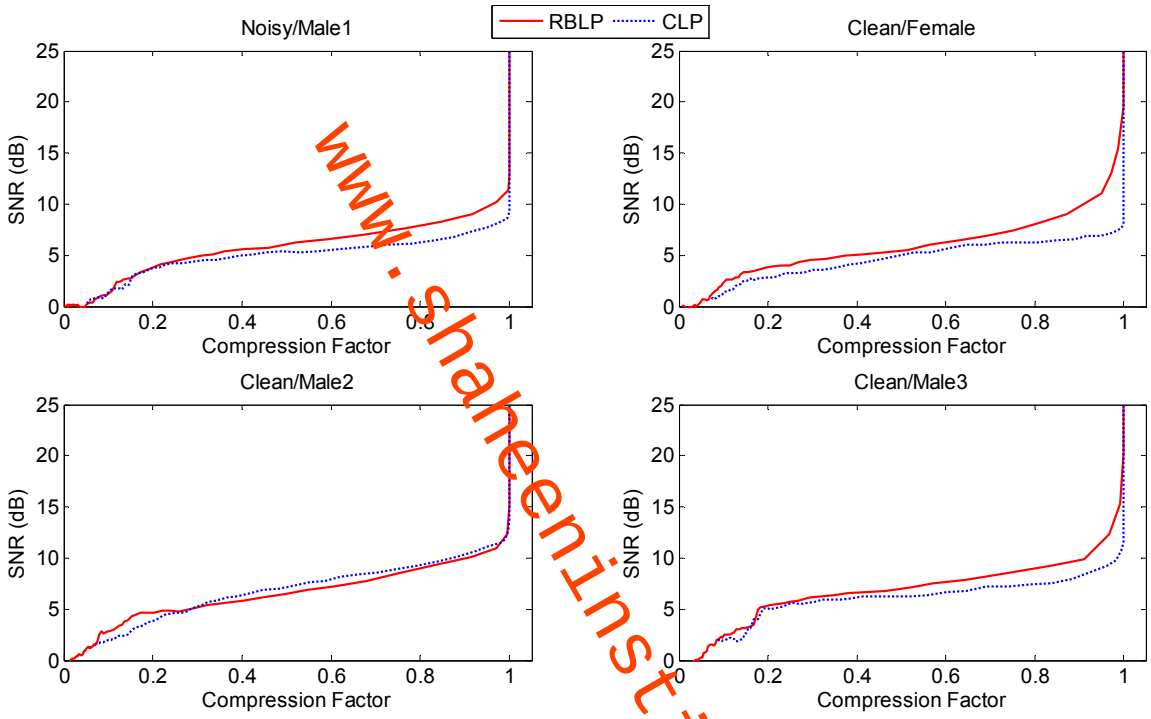

Figure 27. **A comparison between the speech SNR for signals recovered from performing CS on CLP and RBLP obtained residuals.**

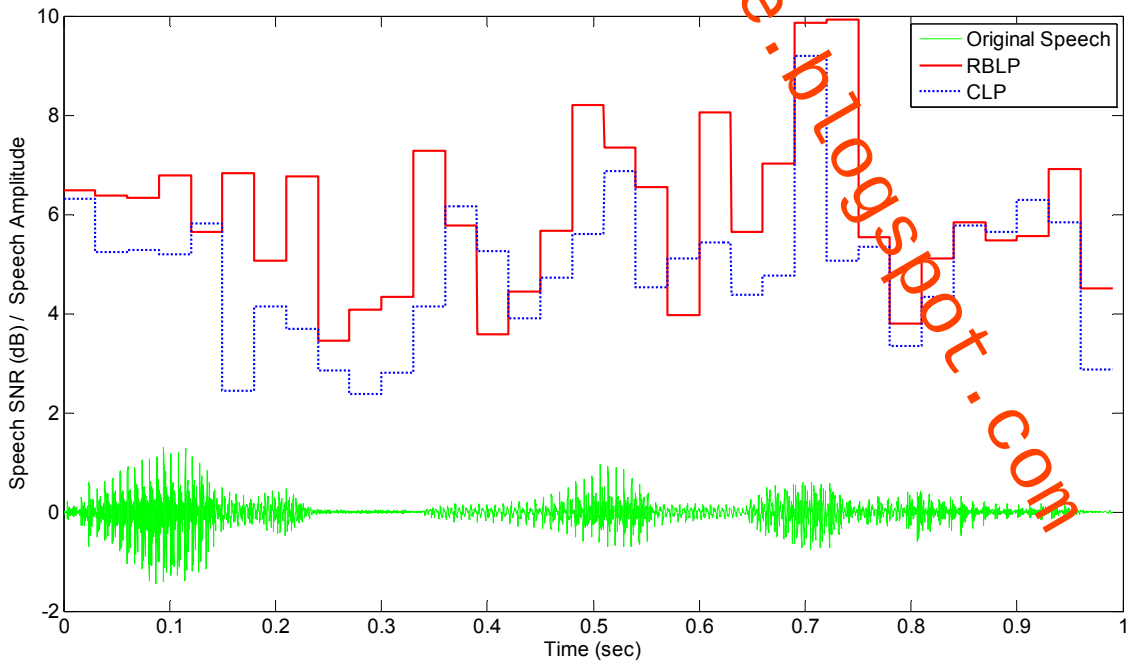

Figure 28. **The original speech signal with the SNR obtained from applying CS at a compression factor of almost 0.4 on both CLP and RBLP residuals for Noisy/Male1 (The speech signal is amplified for the purpose of plotting)**

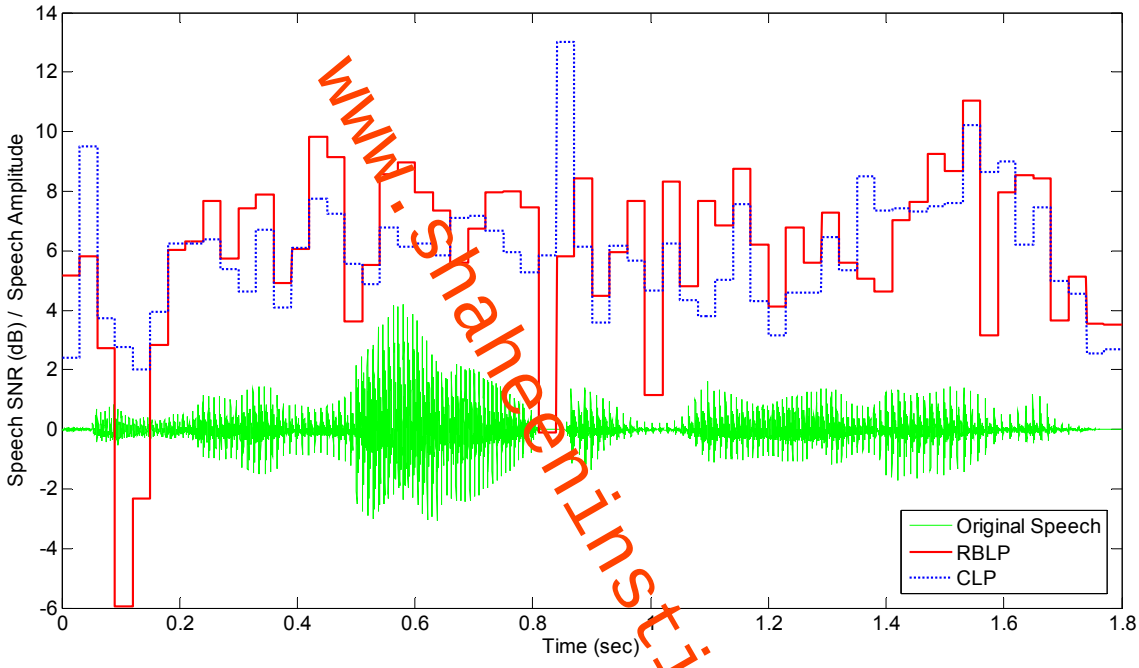

Figure 29. **The original speech signal with the SNR obtained from applying CS at a compression factor of almost 0.4 on both CLP and RBLP residuals for Clean/Male3 (The speech signal is amplified for the purpose of plotting)**

Figure 29 above implies that to get a synthetic speech signal with high SNR one can use RBLP residuals for voiced frames and CLP for unvoiced frames. Such a process requires classifying the speech frame into voiced or unvoiced; which alone is a challenge as described in Section 2.2. However, dealing with error in the unvoiced frames can always be compromised for since theses frames already sound almost like noise. In addition, Figure 29 is just an SNR plot; listening tests confirm that the quality of the speech for the RBLP case is always better than for the CLP.

5.5 FINDING THE BEST THRESHOLD LEVEL

In the previous sections, the performance of the CS process was tested for different speakers over different compression factors. The compression factor was found using equation (5.2) as the summation of all the sensed pulses for all the frames divided by the size of the speech signal; where the number of sensed pulses is found using equation (5.1) by multiplying the frames length logarithm by the number of nonzero pulses (sparsity) of the frame. So far the sparsity of a frame was determined by thresholding the normalized power residuals signal to different levels then counting the number of nonzero pulses left after thresholding. But what is the best thresholding level? To answer this question two different approaches to finding the best threshold, and thus sparsity, are tested.

Referring to Figure 23; thresholding results in having frames with a large number of nonzero pulses and other frames with just a few pulses. And since CS methods cannot recover signals with very few pulses unless the number of measurements is increased, one approach to finding the best, yet the highest, threshold would be to increase the threshold level until no frame has less than 10 pulses, since from Figure 20 at sparsity T=10 the probability that CS fails to recover the data is very small (about 0.5%), this would result in a compression factor that varies from 0.6 to 0.8 depending on the speech signal. For Male3 speaker, the compression factor was about 0.597 with an SNR of 6.66.

Another approach would be to fix the number of pulses in each frame; that is to zero out all but the highest T pulses in each frame. And since it was shown by Makhoul and Berouti [22] that about 88% of the residuals are of very small values, one can keep only the highest 12% pulses of each frame and zero out everything else. For our case each frame is 30 ms long, thus

the highest 12% pulses are the highest 29 pulses. This results in a compression factor of exactly 0.66, where the SNR increases up to 3 dB higher than the previous approach.

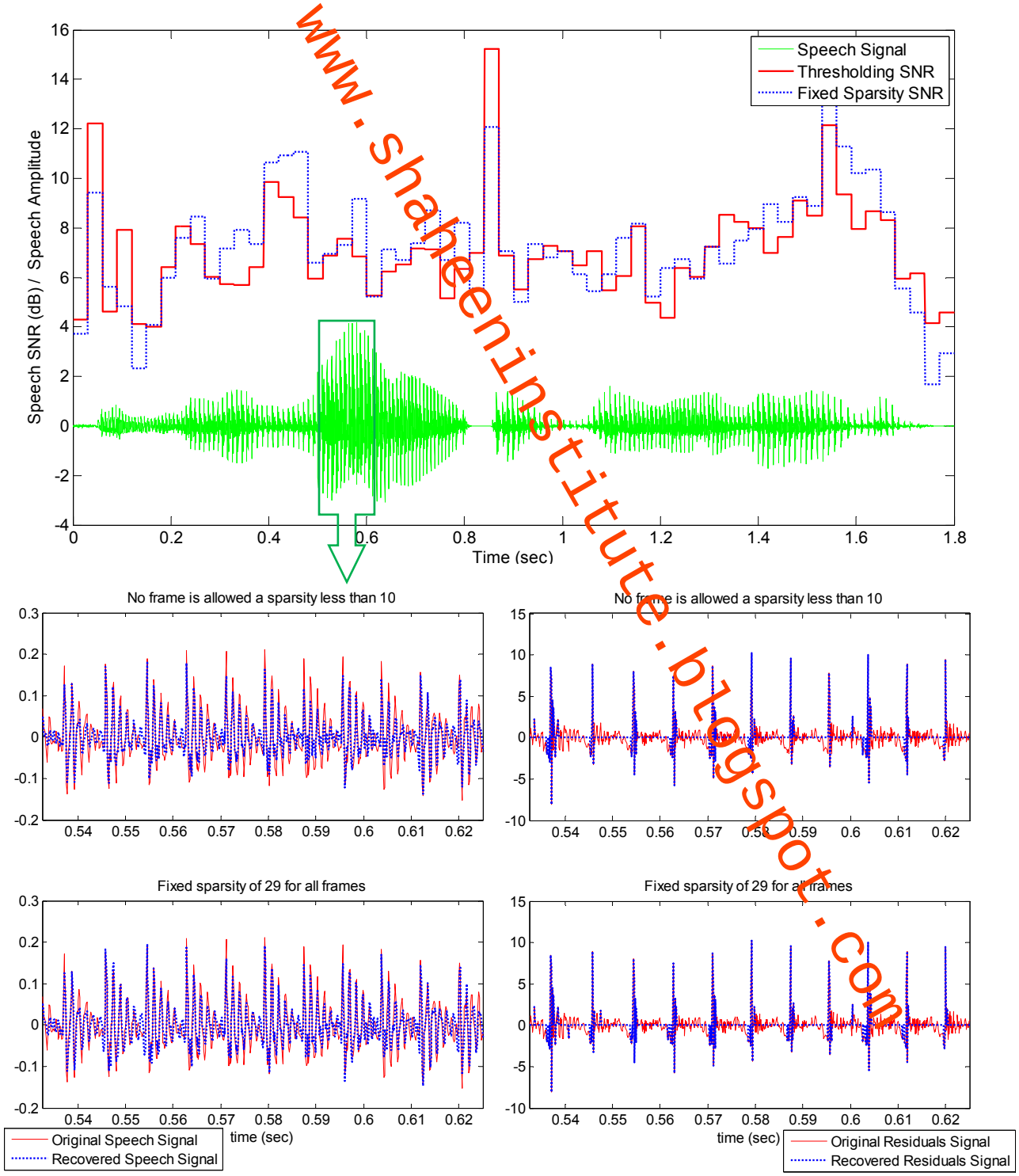

Figure 30. **Recovered residuals and speech signals along with a frame-by-frame speech SNR for Male3 speaker with fixed sparsity and conditioned thresholding.**

Figure 30 compares the two approaches described above for Male3 speaker. The plots show the improvement in the SNR for the voiced frames for the case when the sparsity is fixed for all the frames. However the improvement is not consistent where there are some voiced frames that have higher SNR for the thresholding case. Also it is fair to note that the compression factor for the fixed sparsity case is higher and thus it is expected to have higher SNR. The plots also compare a zoomed in segment of both the residual and the speech for the two approaches.

The plots in Figure 30 show that the reconstruction is fair for both cases, the SNR and quality of the synthesized speech is also high, and it is difficult to favor one approach on the other. However, the conditioned thresholding approach results in compression factors that vary according to the speech signal, which is not preferable.

6.0 SPECTRALLY SHAPING THE CS RECOVERY NOISE

In this chapter, we introduce the spectral shaping of the noise that results from the compressive sampling process. Since we apply CS on the speech residuals then synthesize the speech from the recovered residuals, the error that results (between the original speech and the synthetic one) is similar to the error that results from synthesizing the speech from quantized residuals. The literature [22], [23] reports that the quantization noise can be spectrally shaped and this results in higher SNR and better synthetic speech quality. The spectral shaping method reported in [22] is adapted for the CS noise in order to enhance the quality of the synthesized speech.

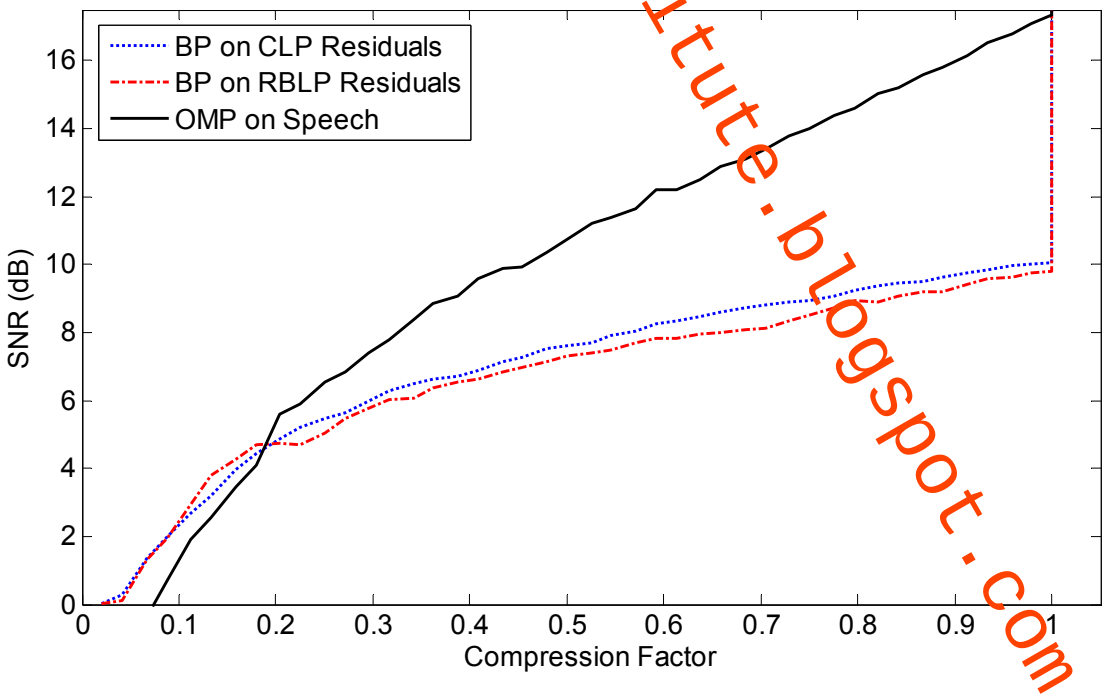

Figure 31. SNR curves for Male2 recovered by CS applied on the residuals and the speech signals

Figure 31 above shows the SNR curves for a speech signal, Male 2, recovered using CS by minimizing the ℓ-1 norm when applied on residuals obtained by CLP and RBLP. Also plotted

in Figure 31 is the SNR curve for the same signal recovered using the orthogonal matching pursuit algorithm (solid line) as presented by Sreenivas and Kleijnin [24]. The work presented in [24] applies CS directly onto the speech signal, not to the residuals like we did in the previous chapter, where the minimization problem is stated as,

$$[\hat{\boldsymbol{e}}, \hat{\mathbf{H}}] = \min_{e,h} \|\boldsymbol{y} - \boldsymbol{\Phi}\,\mathbf{H}\,\boldsymbol{e}\|_2 \quad \text{subject to} \quad \|\boldsymbol{e}\|_0 = T \quad \text{and} \quad \hat{\boldsymbol{s}} = \hat{\mathbf{H}}\,\hat{\boldsymbol{e}} \tag{6.1}$$

The method still uses the residuals as the sparse excitation domain but uses an additional synthesis filter denoted by **H,** therefore the sensed signal is a speech signal rather than a residual. A codebook of size L is constructed of matrices from the training speech data, where $\mathbf{H} \in \{\mathbf{H_1}, \mathbf{H_2}, \ldots, \mathbf{H_L}\}$, and $\boldsymbol{y} = \boldsymbol{\Phi}\,\boldsymbol{s}$.

In order to regenerate the results in [24] we apply CS on raw speech data (with no thresholding) with **H** selected to be a Toeplitz matrix for the linear convolution of the impulse response of $1/A(z)$, instead of construction a codebook. Then the CS problem is solved using the OMP approach explained in Section 4.2.2. The OMP algorithm basically searches for the residual $\boldsymbol{e}$ that minimizes the squared error in the reconstructed speech, under the constraint that $\boldsymbol{e}$ has a sparsity of T. The plot shows a great improvement in the SNR over our approach especially for higher compression factor; however, the resultant speech quality was worse.

The increase in the SNR with the degradation of the speech quality is due to the effect of shaping the noise, as will be explained in Section 6.1. We investigate the effects of noise shaping on the performance of the CS recovered speech signal.

This chapter is divided into two sections. In Section 6.1 we briefly introduce the spectral shaping concept; it is not an attempt to thoroughly explain the theory, the reader is referred to [22], [23] for detailed discussions and derivations. In Section 6.2 we include a noise shaping

stage into our CS process. Various shaping filters are used and the recovered signal is investigated and compared to the original one.

6.1 ADAPTIVE PREDICTIVE CODING AND NOISE SHAPING

Figure 32 below shows two quantization systems; (top) a traditional unit variance quantization system and (bottom) an adaptive predictive coding system used to whiten the noise that results from synthesizing the speech from quantized residuals. From here on; noise refers to the overall system noise (the error between the original speech and the recovered speech) and is different than the quantization noise (the error between the input and output of the quantizer). The quantization noise, $u(n)$ in Figure 32, is defined as $u(n) = \hat{e}(n) - e(n)$.

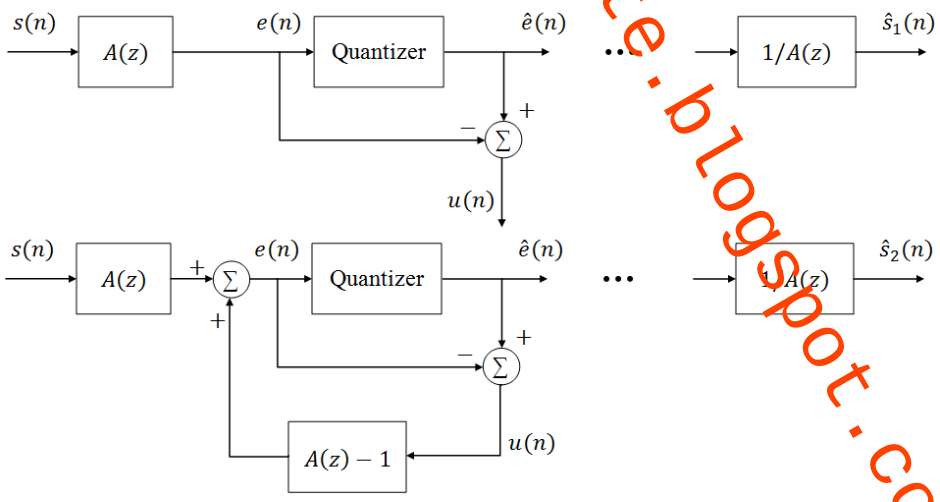

Figure 32. System 1, a block diagram of a traditional quantization system (top) and system 2 a quantization system with adaptive prediction (bottom) [22]

Working in the frequency domain, the output of the system on the top graph is,

$$\hat{S}_1(z) = S(z) + \frac{U(z)}{A(z)} \quad \Rightarrow \quad \text{Noise}_1 = \frac{U(z)}{A(z)} \tag{6.2}$$

In the time domain, the noise is defined as,

$$\text{Noise}_{t1} = u(n) * h(n) \tag{6.3}$$

where $h(n)$ is the impulse response of the inverse filter, $1/A(z)$, and $*$ is a convolution operation.

Thus the power of the noise is,

$$\sigma^2_{Noise_1} = \sigma_u^2 \sigma_h^2 \quad \Rightarrow \quad \text{SNR}_1 = \frac{\sigma_s^2}{\sigma_u^2 \sigma_h^2} \tag{6.4}$$

Equation (6.2) indicates that the noise has the shape of $1/A(z)$, since the quantization error is spectrally flat. Whereas for the bottom graph the noise spectrum is flat and the output is given by

$$\hat{S}_2(z) = S(z) + U(z) \quad \Rightarrow \quad \text{Noise}_2 = U(z) \tag{6.5}$$

And thus the noise power is

$$\sigma^2_{Noise_2} = \sigma_u^2 \quad \Rightarrow \quad \text{SNR}_2 = \frac{\sigma_s^2}{\sigma_u^2} \tag{6.6}$$

Comparing (6.4) and (6.6), SNR_2 is greater than SNR_1, since $\sigma_h^2 > 1$ by definition. Hence whitening the noise increases the SNR. However, the speech is said to suffer from background noise in higher frequencies, causing the speech quality to degrade. It was reported in [22] that further shaping of the noise using a filter $B(z)$ as in Figure 33 increases the speech quality if $B(z)$ is carefully picked such that the noise spectrum falls below the speech spectrum envelope and is almost parallel to it at all frequencies. The output of the system in Figure 33 is given by,

$$\hat{S}_3(z) = S(z) + U(z)B(z) \quad \Rightarrow \quad \text{Noise}_3 = U(z)B(z) \tag{6.7}$$

where Noise_3 has the shape of $B(z)$ since $U(z)$ spectrum is flat.

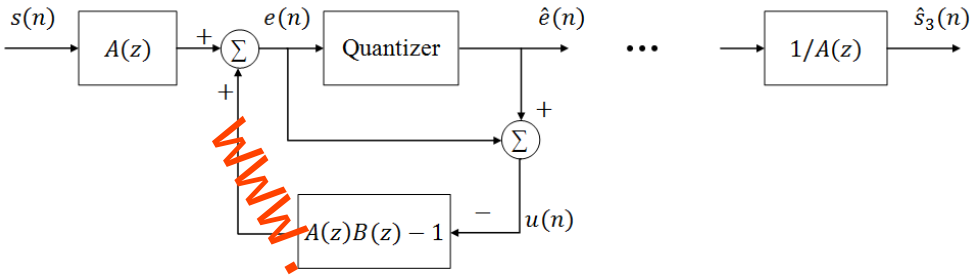

Figure 33. System 3, a block diagram of an adaptive predictive coding system with noise shaping [22]

Therefore, if $B(z) = 1/A(z)$ then system 3 reduces to system 1; and if $B(z) = 1$, system 3 reduces to system 2. In other words, the noise shape of system 3 follows the shape of $B(z)$. In the next section, we modify the CS process to incorporate spectral shaping of the CS noise; several choices of $B(z)$ are tested and the resultant speech is investigated.

6.2 SPECTRALLY SHAPING THE COMPRESSIVE SAMPLING ERROR

As shown in Section 6.1, the shape of the noise spectrum follows $B(z)$ spectrum; if $B(z)$ is chosen to be $1/A(z)$, then no shaping is applied and the output noise spectrum has the shape of the speech spectrum. CS however is a different process than quantization, though there is a similarity in the sense that in both processes the speech is synthesized from a different but similar version of the residuals. Investigation the CS noise (the error between the original speech and the CS recovered speech), it was found that the noise almost has the same shape of the input speech; this means that one can increase the SNR and the speech quality by spectrally shaping that noise so that its spectrum fall below the spectrum of the original signal at all frequencies.

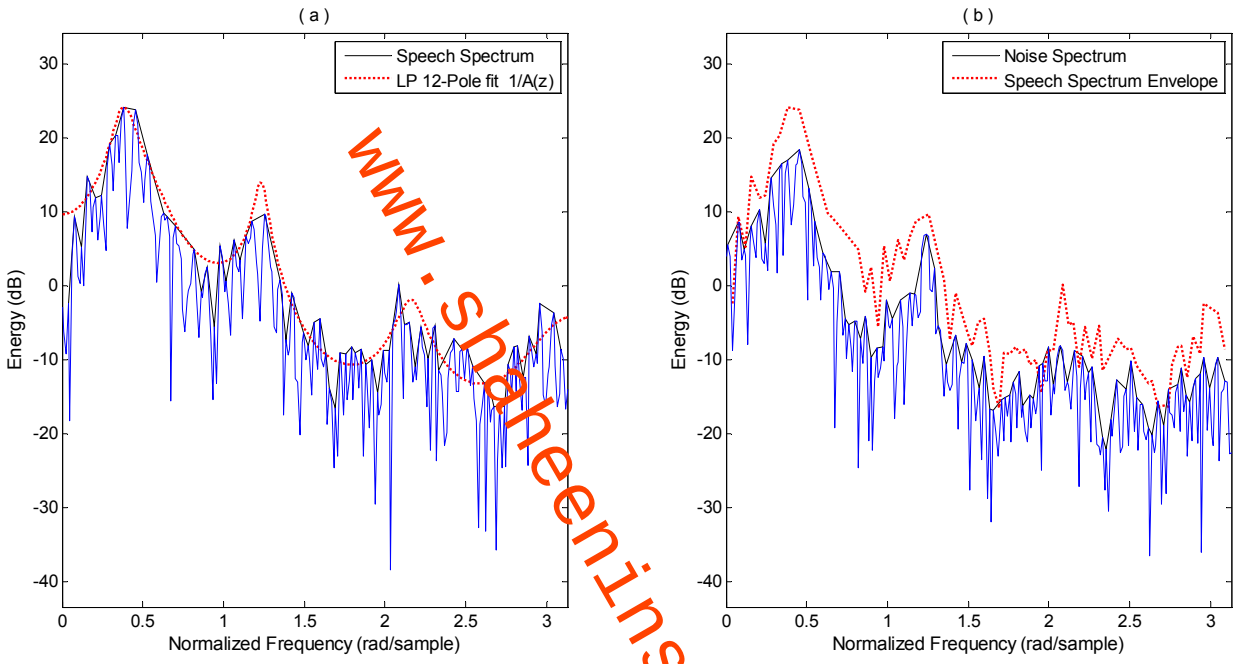

Figure 34. Original speech and CS (on CLP residuals) noise spectrums for Male 2

Figure 34 above shows, (a) the spectrum of a segment of the input speech signal along with the spectrum of the LP inverse filter $1/A(z)$ and (b) the spectrum of the CS noise. It can be seen from the plots in (b) that the CS noise almost has the same shape of the input speech signal. Listening to the noise, it sounds like a whispered version of the speech, which degrades the SNR and results in a recovered signal that suffers from background noise.

Figure 35 below shows the spectrum of the noise that results from applying CS to the speech directly using OMP. The plots, in contrast to the ones in Figure 34 (b), show that the noise is almost white or at least does not have the same shape as the speech signal.

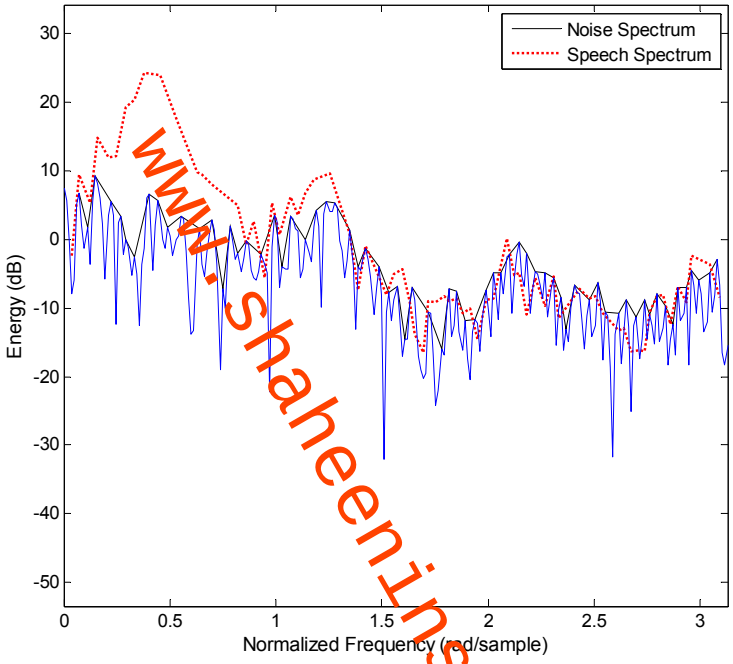

Figure 35. Original speech and CS (on speech using CMP) noise spectrums for Male 2

A question that begs an answer is how to embed a noise shaping step in the CS process? The answer to this question is not obvious, however, the increase in the SNR that results when performing CS on the speech rather than on the residuals implies that the noise was shaped as a result of shaping the sensing matrix.

Let's look at our original CS process, where the sparse representation is the residual basis,

$$\hat{\boldsymbol{e}} = \min_{\hat{\boldsymbol{e}}} \|\hat{\boldsymbol{e}}\|_1 \qquad \text{subject to} \qquad \boldsymbol{y} = \boldsymbol{\Phi}\,\hat{\boldsymbol{e}} \quad \text{and} \quad \hat{\boldsymbol{s}} = \mathbf{H}\,\hat{\boldsymbol{e}} \tag{6.8}$$

Like the quantization error $u(n)$ in Section 6.1, the error in $\boldsymbol{e}$ is spectrally flat. Therefore, synthesizing the speech using $\mathbf{H}$ results in an error, $\boldsymbol{s} - \hat{\boldsymbol{s}}$, that has the spectral shape of $\mathbf{H}$. On the other hand the CS process introduced in [24] is summarized in equation (6.1) as:

$$\left[\hat{\boldsymbol{e}}, \hat{\mathbf{H}}\right] = \min_{e,h} \|\boldsymbol{y} - \boldsymbol{\Phi}\,\mathbf{H}\,\boldsymbol{e}\|_2 \quad \text{subject to} \quad \|\boldsymbol{e}\|_0 = T$$

where, $\hat{\boldsymbol{s}} = \hat{\mathbf{H}}\,\hat{\boldsymbol{e}}$ and $\boldsymbol{y} = \boldsymbol{\Phi}\,\boldsymbol{s} = \boldsymbol{\Phi}\,\mathbf{H}\,\boldsymbol{e}$

But since the noise that results from the process described by equation (6.1) is almost flat, as shown in Figure 35, we argue that one can spectrally shape the noise that results from our original CS process, equation (6.8), by shaping the sensing matrix $\boldsymbol{\Phi}$. The sensing matrix can be shaped by multiplying it with a shaping matrix $\boldsymbol{\Lambda}$ of the impulse response of the shaping filter $\Lambda(\mathrm{z})$. As a result of using $\boldsymbol{\Lambda}$, the error in $\boldsymbol{e}$ is no longer flat but rather has the shape of $1/\boldsymbol{\Lambda}$; and hence the error in $\boldsymbol{s}$ has the spectral shape of $\mathbf{H}/\boldsymbol{\Lambda}$.

Thus the BP minimization problem after incorporating a shaping step becomes:

$$\hat{\boldsymbol{e}} = \min_{\hat{\boldsymbol{e}}} \|\hat{\boldsymbol{e}}\|_1 \qquad \text{subject to} \qquad \boldsymbol{y} = \boldsymbol{\Phi}\boldsymbol{\Lambda}\,\hat{\boldsymbol{e}} \qquad \text{and} \quad \hat{\boldsymbol{s}} = \mathbf{H}\,\hat{\boldsymbol{e}} \tag{6.9}$$

If $\boldsymbol{\Lambda}$ is selected to be 1, no shaping is applied, which results in the CS process as applied in Chapter 5; where the noise has the shape of $1/A(z)$.

Performing CS on the speech signal directly (equivalently, performing CS on the residuals while shaping the sensing matrix with $\boldsymbol{\Lambda} = \mathbf{H}$) to recreate the results presented in [24], using OMP methods for reconstruction, results in an SNR of 9.6 dB and a noise spectrum that is almost flat as shown in Figure 35. But as mentioned above, the synthetic speech sounded worse than the case with no shaping (when BP was used for recovery) even though the SNR is almost 3 dB higher.

Nevertheless, it is understood that BP methods are stronger, though more complicated, than OMP methods; as argued earlier in Section 4.4 and as shown in Figure 18. Therefore, we test shaping the noise that results when applying CS (using BP for recovery) on the CLP residuals with a shaping matrix $\boldsymbol{\Lambda} = \mathbf{H}$. Figure 36 below shows the resulting noise spectrum.

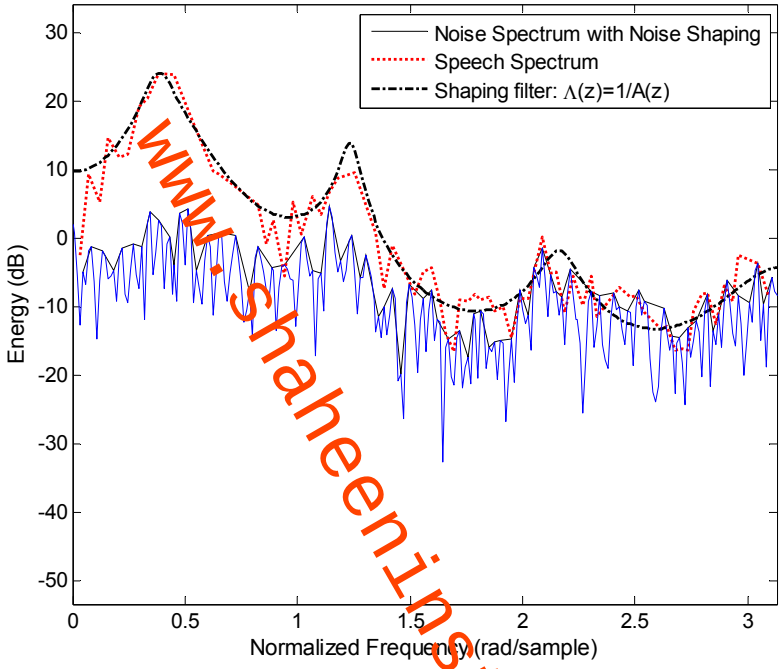

Figure 36. **CS noise spectrum shaped with a filter $\Lambda(z) = 1/A(z)$**

Choosing $\Lambda(z) = 1/A(z)$ results in a CS noise that is almost white, as shown in Figure 36. It also causes the SNR to increase to 12.7 dB (almost 6 dB increase compared to the case of CS with no shaping). Furthermore, the synthesized speech had a very good quality where the roughness that existed before shaping is reduced and with less background noise.

Next, we use a version of the inverse filter whose spectrum has broader peaks as a noise shaping filter to investigate whether the speech quality can be further improved or if the SNR can be increased. Choosing $\Lambda(z) = 1/\sum_{k=0}^{P} a_k \, 0.9^k \, z^{-k}$ where $a_0 = 1$ results in the SNR to go to 10.5 dB and the speech to sound almost the same as the case when $\Lambda(z) = 1/A(z)$ but with added background noise. Figure 37 shows the resulting noise spectrum compared to the original speech spectrum (a) and the frequency response of both the inverse and shaping filters (b).

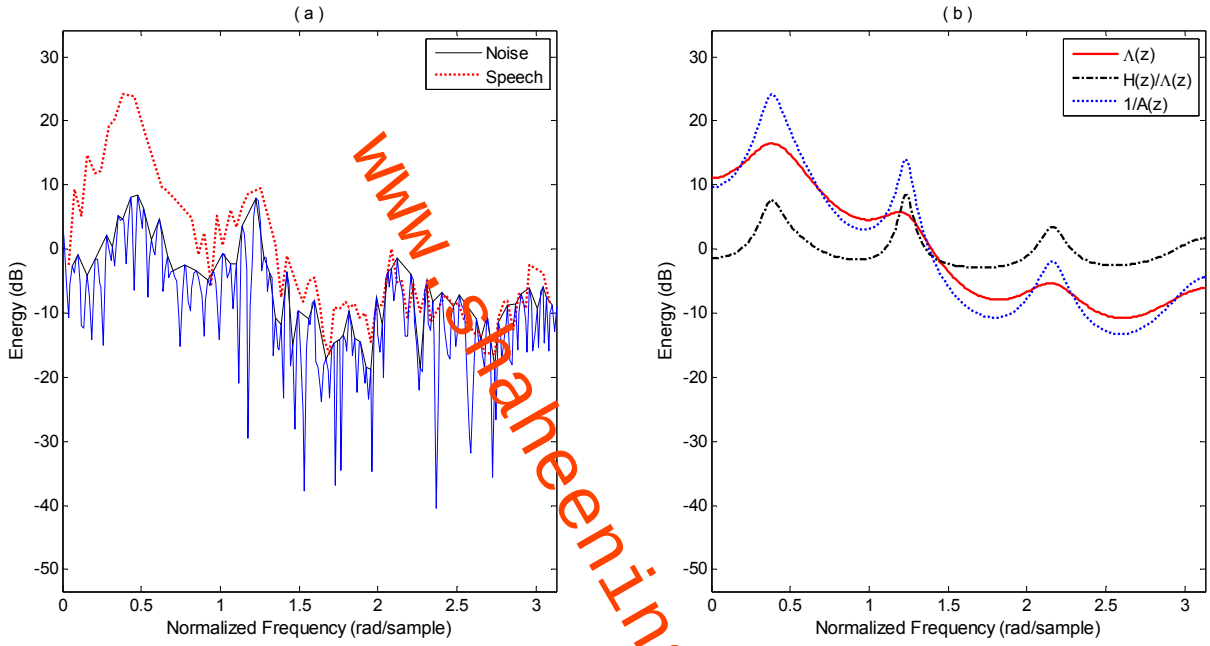

Figure 37. **CS noise spectrum shaped with a filter $\Lambda(z) = 1/\sum_{k=0}^{P} a_k\, 0.9^k\, z^{-k}$**

Figure 37 shows that the noise spectrum is more colored in the formant regions and that justifies the lower SNR; further, the noise spectrum is not well below the speech spectrum for all frequencies, which justifies the lower speech quality.

Referencing Makhoul and Berouti [22], using a shaping filter $B(z)$ that is a second order all-zero fit to the signal spectrum, with the coefficients as in equation (6.10) below, results in a better quality speech since the filter's frequency response lies below the speech spectrum at all frequencies. Thus for the noise to have the shape of $B(z)$, a shaping filter $\Lambda(z) = 1/B(z)A(z)$ is used where the coefficients of $B(z)$ are defined as [22]:

$$b(1) = \frac{\rho_1(\rho_2 - \rho_0)}{{\rho_0}^2 - {\rho_1}^2} \quad , \quad b(2) = \frac{{\rho_1}^2 - \rho_0\rho_2}{{\rho_0}^2 - {\rho_1}^2} \tag{6.10}$$

where $b(0) = 1$, and ρ_i is the i[th] autocorrelation coefficient of $A(z)$ defined as

$$\rho_i = \sum_{k=0}^{p-i} a(k)a(k+1) \tag{6.11}$$

Figure 38 below shows the results of using the shaping filter $\Lambda(z) = 1/B(z)A(z)$. Figure 38 (b) shows that the spectrum of $B(z) = H(z)/\Lambda(z)$ does not fall well below the spectrum of $1/A(z)$; and as a result, the spectrum of the noise is not well below the speech spectrum. Thus the synthetic speech still suffers from hissing noise even though the roughness that existed before shaping is reduced; the resulting speech has an SNR of 10.35 dB.

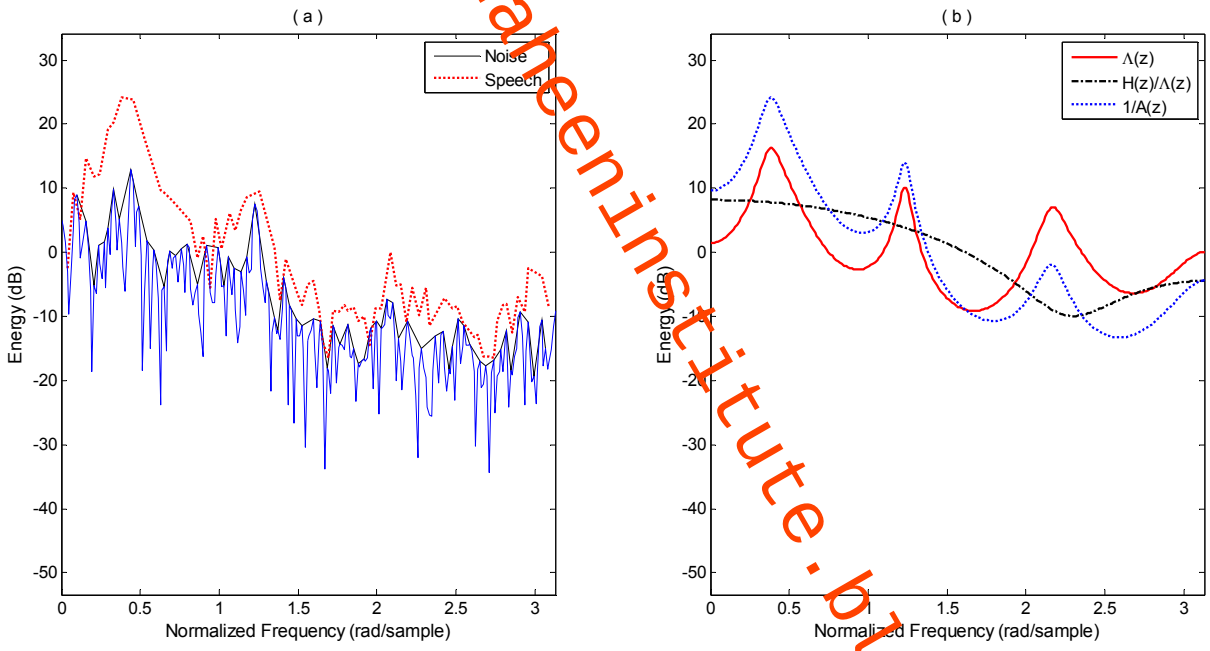

Figure 38. **CS noise spectrum shaped with a filter $\Lambda(z) = 1/B(z)A(z)$**

On the other hand, using the filter $\Lambda(z) = B(z)/A(z)$ results in the noise spectrum to fall well below the speech spectrum at low frequencies and slightly above it at high frequencies. Listening tests show that the speech quality considerably increases where the hissing, roughness and background noise are significantly reduced and the SNR goes to almost 14 dB.

Figure 39 shows the results of shaping with $\Lambda(z) = B(z)/A(z)$; the plots indicate that the low frequency noise is reduced which results in increasing both the SNR and the speech quality.

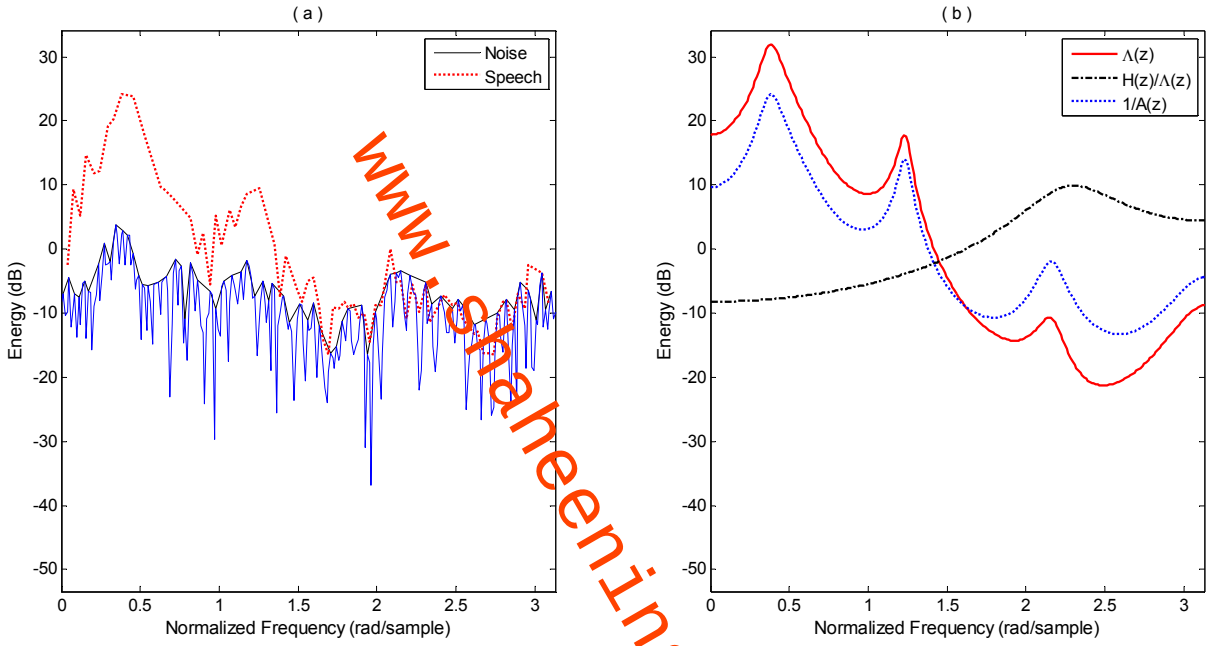

Figure 39. **CS noise spectrum shaped with a filter $\Lambda(z) = B(z)/A(z)$**

Table 1 below summarizes the results of embedding a noise shaping step into the CS process.

Table 1. **Noise shaping effect on the CS/CLP recovered speech for Male2 at compression factor of 0.4**

Filter transfer function	Speech SNR	Effect on speech quality
No shaping $\Lambda(z) = 1$	6.9 dB	Speech suffers from roughness and sounds synthetic.
Sreenivas and Kleijnin [24], $\Lambda(z) = 1/A(z)$ with OMP	9.6 dB	Speech suffers from roughness, hissing and high background noise.
$\Lambda(z) = 1/A(z)$	12.7 dB	Good speech quality, and the roughness is noticeably reduced.
$\Lambda(z) = \frac{1}{\sum_{k=0}^{P} a_k \, 0.9^k \, z^{-k}} \;, a_0 = 1$	10.5 dB	Good speech quality; and the roughness is noticeably reduced.
$\Lambda(z) = \frac{1}{B(z)A(z)}$	10.35 dB	Good speech quality; the roughness is reduced but with high background noise.
$\Lambda(z) = \frac{B(z)}{A(z)}$	13.9 dB	Very good speech quality; and the roughness is drastically reduced.

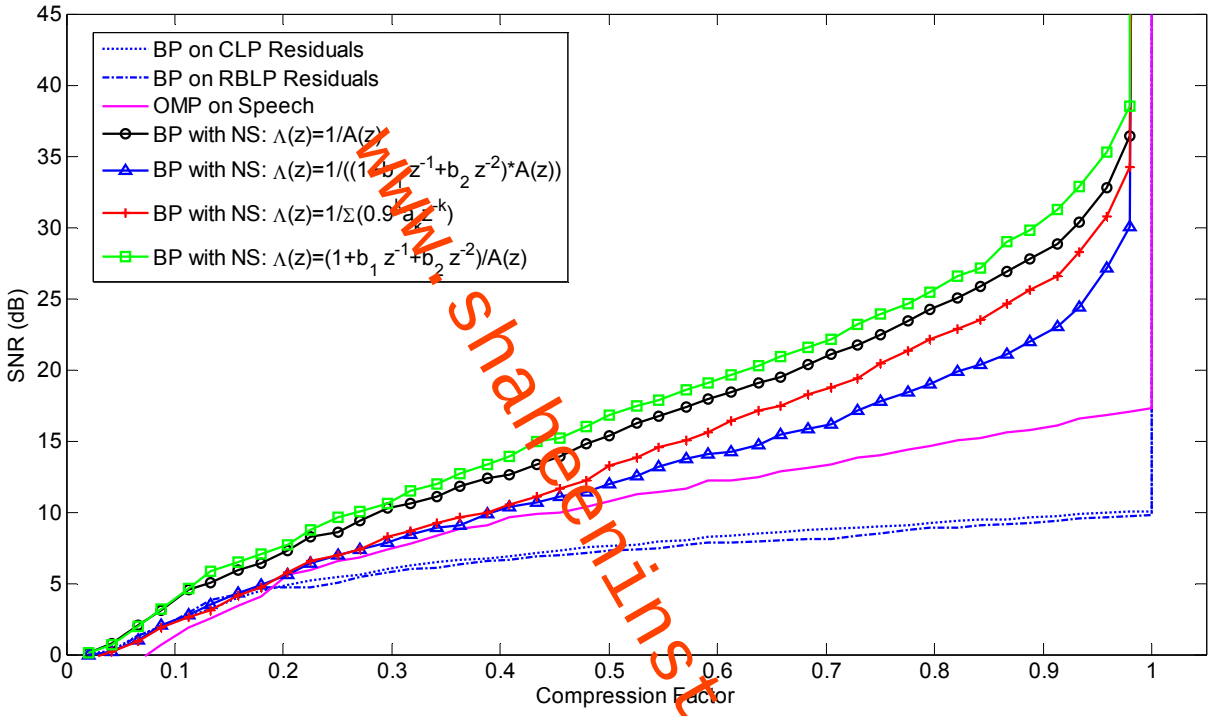

Figure 40. **SNR curves for Male2 for speech recovered using CS on CLP residuals with and without noise shaping**

Figure 40 shows the SNR plots for the speech recovered with no shaping by CS on CLP and RBLP residuals, speech recovered by applying CS directly on the speech using OMP, and speech recovered by CS applied to CLP residuals while shaping the noise with various shaping filters.

The results in Table 1 and Figure 40 are for the case where the residuals are obtained from CLP. However, as Chapter 5 recommends, speech recovered by applying CS to the residuals obtained from RBLP has a better quality. Therefore, the spectral shape of CS noise is studied for RBLP residuals; where not only the residuals are of the RBLP but also the inverse filter coefficients that are used to shape the noise are the RBLP filter coefficients.

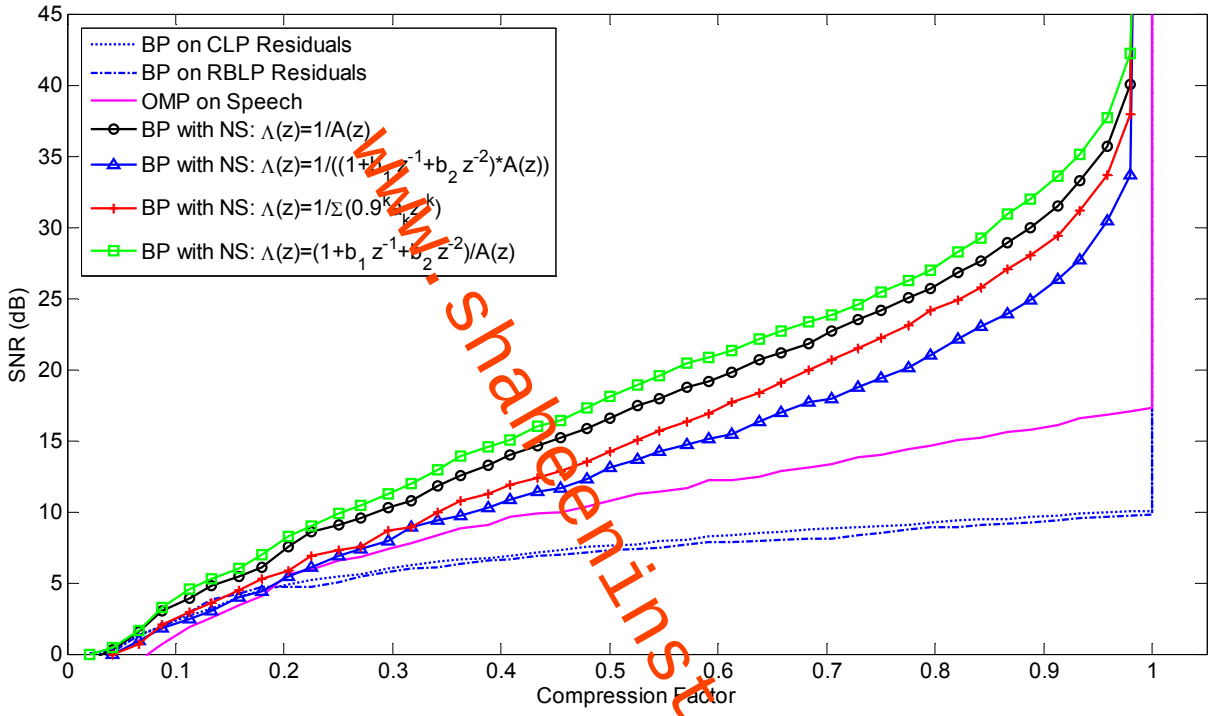

Figure 41. **SNR curves for Male2 for speech recovered using CS on RBLP residuals with and without noise shaping**

Table 2. **Noise shaping effect on the CS/RBLP recovered speech for Male2 at compression factor of 0.4**

Filter transfer function	Speech SNR	Effect on speech quality
No shaping $\Lambda(z) = 1$	6.6 dB	Speech is of a good quality but suffers from roughness and sounds synthetic.
Sreenivas and Kleijnin [24], $\Lambda(z) = 1/A(z)$ with OMP	9.6 dB	Speech suffers from roughness, hissing and high background noise.
$\Lambda(z) = 1/A(z)$	14 dB	Good speech quality; and the roughness is noticeably reduced.
$\Lambda(z) = \frac{1}{\sum_{k=0}^{P} a_k \, 0.9^k \, z^{-k}}$, $a_0 = 1$	11.9 dB	Good speech quality; and the roughness is noticeably reduced.
$\Lambda(z) = \frac{1}{B(z)A(z)}$	10.83 dB	The roughness is reduced but with high background noise.
$\Lambda(z) = \frac{B(z)}{A(z)}$	15.1 dB	Very good speech quality; and the roughness is drastically reduced.

Table 2 and Figure 41 summarize the implementation results and show similar results to those of Table 1 and Figure 40 but for the RBLP case; they confirm that spectral shaping of the noise increases both the SNR and the synthesized speech quality. They further confirm that applying CS to BRLP residuals results in better results than when CS is applied to CLP residuals even in the case where the noise is spectrally shaped.

7.0 SUMMARY OF RESULTS

Speech was successfully compressively sampled as stated in Chapter 5 and Chapter 6.

It was shown in Chapter 5 that speech synthesized from recovering compressed sensed residuals obtained by RBLP had a better quality than speech synthesized from CLP CS recovered residuals; which makes sense since the robust linear prediction filter provides a better fit to the speech spectrum. Even though for some speakers, e.g. Male 2, the SNR is higher for the CLP case; the quality of the speech sounded better for the RBLP case. It was also shown that setting a fixed sparsity level for all the residuals frames yields a fixed compression factor that is know at the beginning of the process; which is better than setting a threshold level where the compression factor cannot be determined until the threshold step is done. Also, setting a fixed sparsity level guarantees that no frame is left with a low number of pulses. It was shown in Chapter 4, Figure 20, that signals with very few pulses are more likely not to be successfully recovered unless more measurements are taken. For the case of a signal with length 240, it was shown that for example signals with only 4 pulses are likely not to be recovered almost 54% of the time compared to signals with 12 pulses, for example, that are likely to be successfully recovered almost 90% of the time.

The results from Chapter 6 show that the speech quality and the SNR can be well improved by spectrally shaping the CS noise. For all the tested shaping filters, no more parameters are needed to be transmitted since the shaping filters coefficients can be calculated at the receiver end using the LPC's. Implementation results were compared to the results of a

previous work on compressive sampling of speech [24] where the CS process is performed directly on the speech signals and OMP algorithm are used for recovery.

As seen in Figure 40, the SNR curves for the case when compressive sampling is performed by BP methods are much higher than when the case where OMP methods are used. This is expected since, as shown in Section 4.2 and Figure 18, BP algorithms are stronger than OMP algorithms. However, BP algorithms are far more complex. It was reported [21] that the OMP algorithm described in Section 4.2 has a complexity of $\mathcal{O}(TmN)$ compared to $\mathcal{O}(m^2N^{3/2})$ for some BP approaches, which indicates that the complexity grows quadratically with the number of measurements. A simple indication of the complexity of the BP in comparison with OMP is the execution time. For example, recovering a speech signal of 2.64 sec, Male 2, at a compression factor of 0.4 using OMP is executed in almost 0.6 sec, while performing the same operation using BP is executed in about 19 sec. However, one can always compromise between the quality of the recovered speech and the complexity of the process.

CONCLUSION

Compressive sampling offers an acquisition scheme that is very simple at the transmission end, where the signal is sampled using a random sensing matrix that acquires a number of measurements less than the dimension of the signal. On the other hand, at the receiver end, the process is quite complex and is expensive in terms of computation when BP algorithms are used.

We apply CS on speech residuals and conclude that it is more efficient to apply compressive sampling on residuals obtained from robust linear prediction algorithms than the conventional linear prediction residuals.

We show that the noise that results from CS can be spectrally shaped using shaping filters with coefficients that can be calculated from the linear prediction filter's coefficients. It was shown that spectrally shaping the noise improved the SNR, minimized the effect the CS noise and resulted in high quality speech with high SNR up to 15 dB at a compression factor of 0.4.

FUTURE WORK

Applying compressive sampling to speech residuals obtained from robust linear prediction analysis then spectrally shaping the resultant noise using a pole/zero filter results in satisfying results for a compression factor of 0.4 and above. A compression factor of 0.4 was achieved when keeping the largest 18 pulses in the residuals signal then taking $18\,C\log(N)$ measurements where C was set to 1. However, smaller C's that generate smaller compression factor haven't been studied. The plots in Figure (20) point out that for the same signal length, smaller C's can be used if the sparsity is higher. Hence, one can compromise the sparsity of the residuals by the number of measurements taken (higher C) which may generate interesting results.

The sensed signal will be quantized then sent over the transmission channel where more noise is added. If the compressed sensed signals are very sensitive to channel or/and quantization noise, CS cannot be used as a practical acquisition scheme. Therefore, it is important to study the effects of quantization on the sensed signals. Although this is beyond the goal of this research, we studied the sensed signals and found that they are normally distributed with almost zero mean and unit variance. However, the effects of quantization and channel noise still need to be studied.

APPENDIX A

SPEAKERS AND SENTENCES INFORMATION

Noisy/Male1: “Alice in wonderland”; recorded using a commercial microphone.

Clean/Female: “I ate every oyster on Nora's plate”; from the TIMIT database.

Clean/Male2: “Don't ask me to carry an oily rag like that”; from the TIMIT database.

Clean/Male3: “He will allow a rare lie”; from the TIMIT database.

APPENDIX B

EXTENDED RESULTS FOR DIFFERENT SPEAKERS

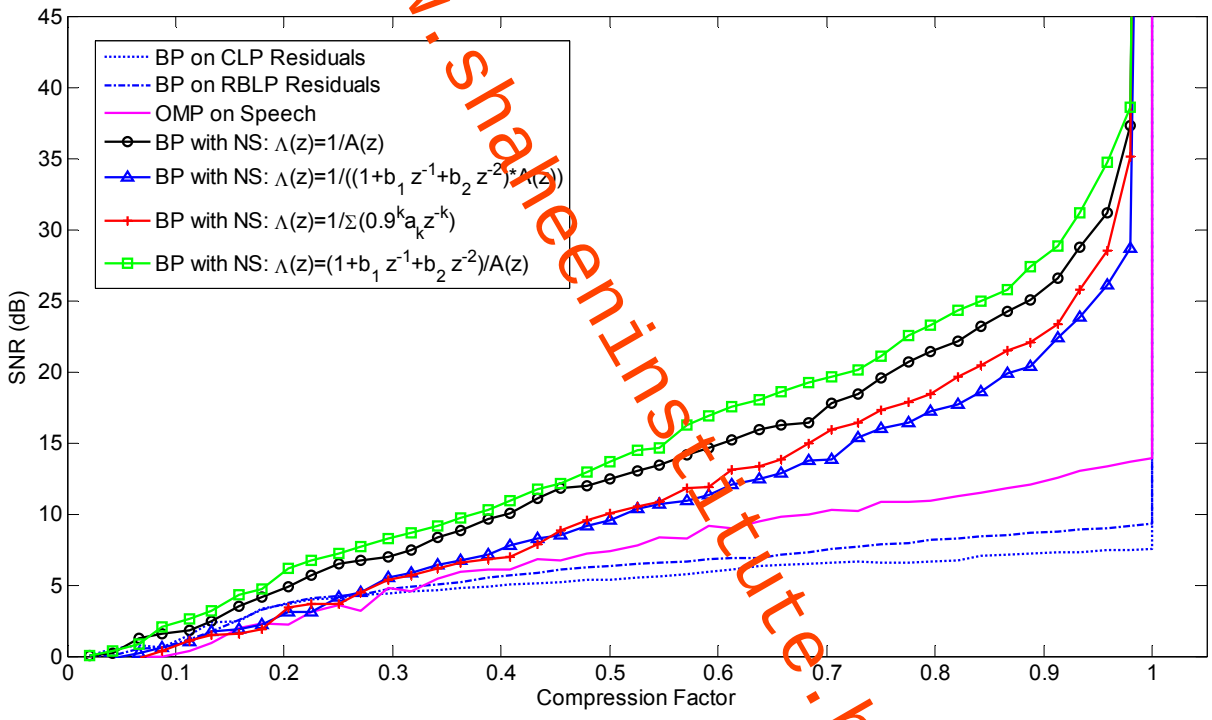

Figure 42. **SNR curves for Male1/Noisy for speech recovered using CS on RBLP residuals with and without noise shaping**

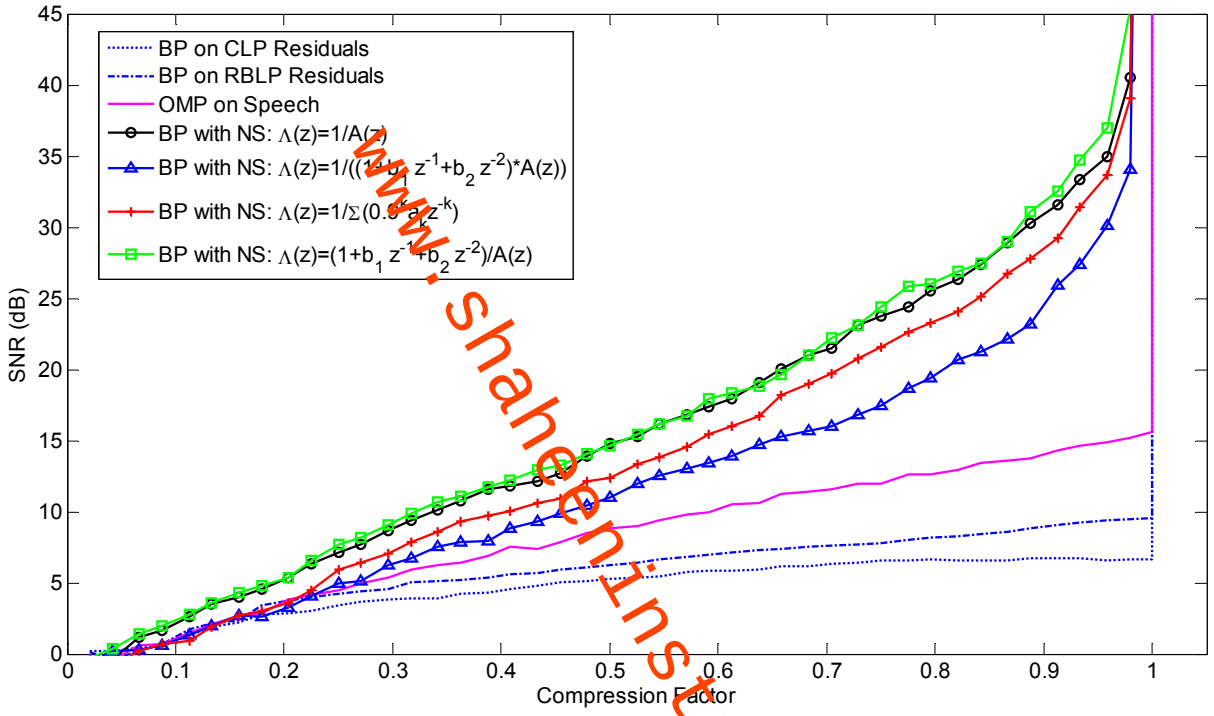

Figure 43. **SNR curves for a Female Speaker for speech recovered using CS on RBLP residuals with and without noise shaping**

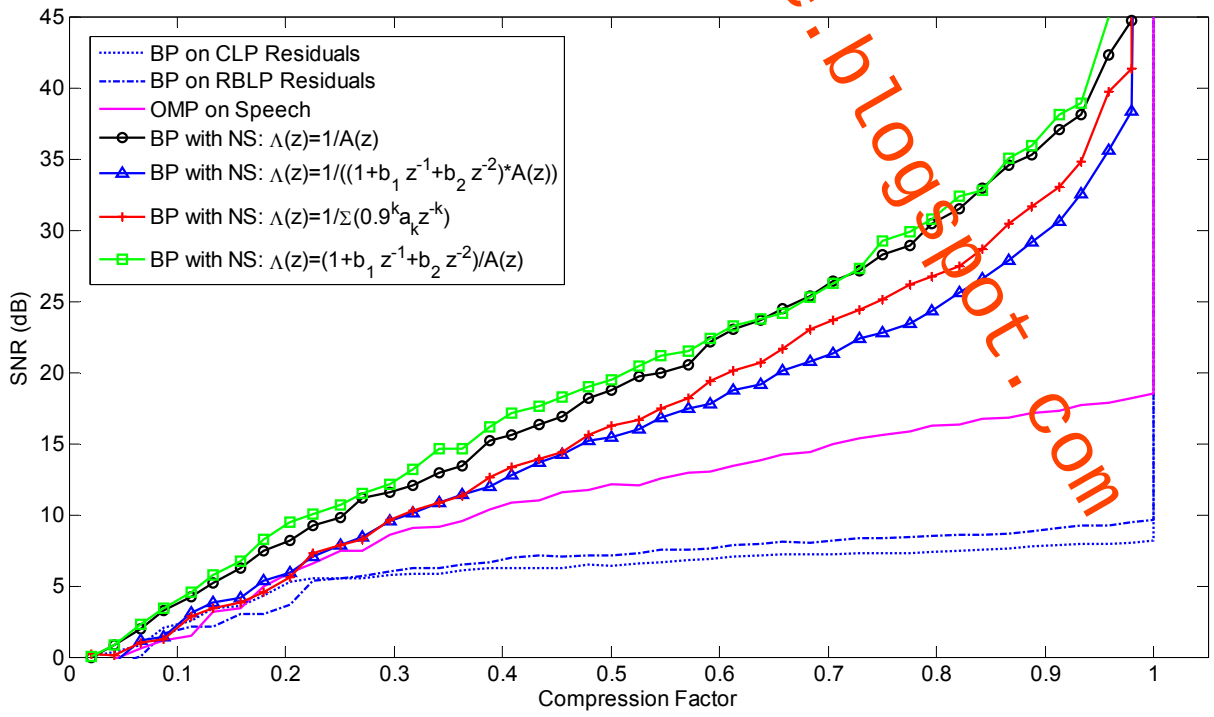

Figure 44. **SNR curves for Male3 for speech recovered using CS on RBLP residuals with and without noise shaping**

BIBLIOGRAPHY

[1] W. C. Chu, *Speech Coding Algorithms: Foundation and Evolution of Standardized Coders*, New Jersey: John Wiley & Sons, 2003.

[2] Thomas F. Quatieri, *Discrete-Time Speech Signal Processing: Principles and Practice*, New Jersey: Prentice Hall, 2001.

[3] L. R. Rabiner and R. W. Schaffer, *Digital Processing of Speech Signals*, New Jersy: Prentice Hall, 1978.

[4] A. M. Kondoz, *Digital Speech: Coding For Low Bit Rate Communication Systems*, Chichester: John Wiley & Sons, 2004.

[5] B. S. Atal and J. R. Remde, "A new model of LPC excitation for producing natural-sounding speech at low bit rates," *Proc. IEEE Int. Conf. Acoust., Speech, and Signal Processing*, Vol. 7, pp. 614–617, 1982.

[6] J. Makhoul, "Linear prediction: a tutorial review," *Proc. IEEE*, Vol. 63, NO. 4, pp. 561–580, April 1975.

[7] R.Viswanathan and J. Makhoul, "Quantization properties of transmission parameters in linear predictive systems," *IEEE Trans., Acoust., Speech, and Signal Processing*, Vol. NO. 3, pp. 309–321, June 1975.

[8] K. Ozawa, S. Ono, and T. Araseki, "A study on pulse search algorithms for multipulse excited speech coder realization," *IEEE J. Select. Areas Commun.*, Vol. SAC-4, NO. 1, pp. 133–141, January 1986.urnal.AC

[9] S. Singhal and B. S. Atal, "Improving performance of multi-pulse LPC coders at low bit rates," *IEEE Int. Conf. Acoust., Speech, and Signal Processing*, Vol. 9, NO. 1, pp. 9–12, 1984.

[10] J. Makhoul , "Spectral analysis of speech by linear prediction," *IEEE Trans. Audio Electroacoust.*, Vol. 21, NO. 2, pp. 140–148, June 1973.

[11] R. P. Ramachandran, M. S. Zilovic, and R. J. Mammone, "A comparative study of robust linear predictive analysis methods with applications to speaker identification," *IEEE Trans. Speech, and Audio Signal Processing*, Vol. 3, pp. 117–125, 1995.

[12] C. H. Lee, "On robust linear prediction of speech," *IEEE Trans. Acoust., Speech, and Signal Processing*, Vol. 36, NO. 5, pp. 642–650, May 1988.

[13] David G. Luenberger and Yinyu Ye, Linear and Nonlinear Programming, Springer, 2008.

[14] D. L. Donoho, "Compressed sensing," *IEEE Trans. Inform. Theory*, Vol. 52, NO. 4, pp. 1289–1306, April 2006.

[15] E. J. Candès and M. B. Wakin, "An introduction to compressive sampling," *IEEE Signal Processing Magazine*, Vol. 25, NO. 2, pp. 21–30, March 2008.

[16] E. Candès and J. Romberg, "Sparsity and incoherence in compressive sampling," *Inverse Problems*, Vol. 23, NO. 3, pp. 969–985, 2007.

[17] J. Li, M. Gabbouj, J. Takala, and H. Chen, "Laplacian modeling of DCT coefficients for real-time encoding," *IEEE Int. Conf. Digital Object Identifier*, pp. 797–800, 2008.

[18] E. J. Candès, *Talks at the University of Minnesota*, Compressive Sampling and Frontiers in Signal Processing, June 2007. [Online] [Cited: 11 20, 2009.] http://www.ima.umn.edu/2006-2007/ND6.4-15.07/abstracts.html

[19] T. T. Do, T. D. Tran, and Lu Gan, "Fast compressive sampling with structurally random matrices," *IEEE Int. Conf. Acoust. Speech and Signal Processing*, pp. 3369–3372, 2008.

[20] E. Candès, J. Romberg, and T. Tao, "Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information," *IEEE Trans. Inform. Theory*, Vol. 52, NO. 2, pp. 489–509, 2006.

[21] J. A. Tropp and A. C. Gilbert, "Signal recovery from random measurements via orthogonal matching pursuit," *IEEE Trans. Inform. Theory*, Vol. 53, NO. 12, pp. 4655–4666, 2007.

[22] J. Makhoul and M. Berouti, "Adaptive noise spectral shaping and entropy coding in predictive coding of speech," *IEEE Trans. Acoust., Speech, and Signal Processing*, Vol. 27, NO 1, pp. 63–73, 1979.

[23] B. S. Atal and M. R. Schroeder, "Improved quantizer for adaptive predictive coding of speech signals at low bit rates," *IEEE Int. Conf. Acoust. Speech and Signal Processing*, pp. 535–538, 1980.

[24] T.V. and W. B. Kleijn, "Compressive sensing for sparsely excited speech signals," *IEEE Int. Conf. Acoust. Speech and Signal Processing*, pp. 4125–4128, 2009.